自動搬送制御を組み込んだ
医用分析装置プラットフォーム開発に関する研究

兒玉　隆一郎

電気通信大学　大学院　情報システム学研究科
情報メディアシステム学専攻

博士（工学）の学位申請論文

2015年3月

自動搬送制御を組み込んだ
医用分析装置プラットフォーム開発に関する研究

電気通信大学　大学院　情報システム学研究科

情報メディアシステム学専攻

博士（工学）の学位申請論文

審査委員会

主査　田野　俊一　教授

委員　阪口　豊　　教授

委員　末廣　尚士　教授

委員　古賀　久志　准教授

委員　田原　康之　准教授

The Development of Clinical Laboratory Instrument Software Platform Embedding Automatic Transportation Control

Ryuichiro Kodama

Abstract

A wide variety of clinical laboratory tests are conducted supporting medical diagnosis. The testing is automated in the laboratory test field for specimen such as blood. The laboratory testing has been automated by individual instruments that add reagent to specimen and calculate test data based on the chemical reaction. While automation of testing by instruments was being realized, laboratories are adopting the system in which multiple instruments connected to conveyors enhance the test processing performance. The specimen is transported to designated instruments by conveyors. The system requires software platform to connect instruments to conveyors. It is important to design system performance corresponding to connected instruments and to apply a software development method to maintain platform containing unchanged core assets.

This study is a "system development thesis" to report how to realize the integrated software platform embedding automatic transportation control for specimens. This thesis is composed of (1) the design of automatic transportation control to enhance system processing performance, and (2) the development method of integration software platform to allow flexible connection of instruments to the system.

(1) The design of automatic transportation control

This theme is to propose the basic design of automatic transportation control and conduct quantitative performance analysis for the processing time of clinical instrument system connected by a conveyor. PLS (Pipe Line System) is one of the architectures to implement the system where clinical instruments are aligned along the conveyor and each instrument pipettes samples on the conveyor. PLS was used to deal with almost uniformly requested tests for samples in the 1990s. After that, as a variety of tests is made wide and requested tests are deviated in sample by sample, STS (Side Track System) was devised and is now working. STS has a buffer function and passing function for samples to skip unrequested instrument. STS has its potential capacity in processing performance so that it is of great significance to pursue its possibility. This study clarifies the following: 1) The performance significantly diminishes when the conveyance time per one specimen for a conveyor exceeds (the processing time of a specimen)/ ((number of the devices) + 1) in STS, 2) the processing time of STS is 30% shorter than the one of PLS in average where both systems process the same 200 samples which drop in

four instruments randomly and those instruments have one buffer in each of entrance and exit of the side track, and 3) the increase of buffers does not have impact more than 30% performed by one buffer. Our performance analysis focuses the sequence of random specimen samples which skip a specific instrument, and placed STS as the architecture to compress the space caused by skipping. This study confirms possibility of STS quantitatively and in this way builds the design guideline for the conveyor and the buffer of STS.

(2) The development method of integration software platform

This study proposes the software development method based on Software Product Line (SPL) approach employed for Analyzer Integration Management Software (AIMS) to systemize heterogeneous clinical analyzers. It is difficult to make a development plan to connect a new analyzer to AIMS because an analyzer requires its own particular management and various portion of AIMS software should be changed to implement the new management. To solve this problem, the study devised the method called Architecture Domain Matrix (ADM) method in which each architecture component is further decomposed into clinical operation flow elements and core asset of software is extracted from those elements. This method controls development cost of core asset in a cost estimate phase and enhances productivity of software development because Work Breakdown Structure (WBS) can be generated by collecting all change specifications for each operational flow element and a development team suitable for change can be designed by adding up all changes for each architecture component. After applying this method to a real project, the project integrated embedded software of three different analyzers in one year and a half and achieved 2.5 times embedded software productivity compared with the past non-SPL methods.

The technologies described in the above (1) and (2) are applied to the real clinical instruments. Both (1) and (2) contribute to the development of clinical instruments which occupy a high share of the world market.

The configuration of each chapter is as follows:

Chapter 1 describes the background and purpose of this study.

Chapter 2 builds the problem scope for this study which is composed of the transportation performance design and the platform development method.

Chapter 3 shows how to control the specimen transportation for STS, and clarifies that STS can overcome the PLS deficiency caused by deviation of instrument paths which each specimen follows.

Chapter 4 compares STS and PLS in processing performance in an analytical manner. First, the limit performance of a conveyor to keep STS system performance is analyzed. Second, probability distribution of idle time sequence length is defined so that a transportation system is modeled as a filter transforming that distribution. This clarifies the mechanism for STS to increase the processing performance by reducing idle time.

Chapter 5 verifies the analytical results described in Chapter 4. It verifies that

the limit performance of a conveyor does exist, the filter model of transportation system can explain the idle time reduction in digit, and STS is superior to the old system by calculating compression rate of makespan. In addition, it reports and discusses how the number of instruments and buffers affects the system performance quantitatively.

Chapter 6 proposes how to develop STS software platform. The method is called ADM (Architecture Domain Matrix) method. The problem to be solved by ADM is that every time a new instrument is connected to the platform, the source code changes are scattered everywhere in the architecture. ADM allows achieving higher resolution to analyze source codes by two axes, architecture elements and domain elements. This enhances estimate precision and ADM can appropriately control the development of core assets.

Chapter 7 verifies the high productivity achieved by ADM method and reviews the result after ADM method is applied to a real project.

Chapter 8 describes the real products which the transportation performance design and the platform development method are applied to. STS, which was devised in 1990s, not only enables customers to provide a variety of instrument configurations, but also characterizes the STS products as the system which can avoid unnecessary congestion of samples on a conveyor. This leads to the high share of clinical instruments in the world market. Also the software platform was developed by adopting the ADM method so that many varieties of clinical instruments were built with high productivity. This also contributes to the high world-wide share of clinical instruments.

Chapter 9 summarizes the problems set in the transportation performance design and the platform development method and their solutions described in the above chapters so that those solutions contribute to the world-wide share of the system products.

As the result of this study, the system processing performance and the development productivity has been enhanced establishing how to build the software platform of clinical instrument system from a design phase to an implementation phase. The system products which adopt the methods in this study are utilized in the world to contribute to the cost reduction and the testing speed in clinical laboratories.

自動搬送制御を組み込んだ
医用分析装置プラットフォーム開発に関する研究

兒玉　隆一郎

概要

医療を支える臨床検査部門では様々な検査が行われる．中でも人体から採取された，血液などの検体を検査する分野を検体検査と呼び，自動化が進んでいる．検体検査は検体に試薬を添加し検査項目濃度を自動算出する臨床検査自動分析装置を中心に自動化されてきた．このような分析装置単体による処理性能の追求と並行して，複数の分析装置を搬送路に接続し処理性能を向上させるシステムが開発された．検体は複数の分析装置を渡り必要な検査を実施する．このようなシステムの開発においては複数の分析装置を搬送路に接続するためのプラットフォームが必要となる．その開発にあたっては，接続される装置に対応したシステム処理性能設計とプラットフォームとしての不変なソフトコア資産を維持するためのソフト開発手法が重要となる．

本論文は検体検査における自動搬送制御を組み込んだ統合ソフトウェアシステムを実現する手法について報告する“システム開発型論文”である．本研究は，(1) 処理性能を向上させる自動搬送制御の設計，及び，(2) フレキシブルに分析装置を接続できるプラットフォームの開発手法から構成される．

（１）搬送制御設計

自動搬送制御を組み込んだプラットフォームには，処理性能が異なる分析装置を柔軟に搬送路に接続でき，その稼働率を最大限に引き出す仕組みが求められる．同プラットフォームを構築する方式の 1 つとして，搬送路沿いに並んだ分析装置が検体を吸い取る方式(PLS: Pipe Line System)がある．PLS では，検体を容器から吸い取って反応容器に移すというピペット動作が一番長い装置を待って，搬送路が 1 装置分シフトして次のピペット動作が行われる．PLS は，検査項目のばらつきが少ない検体を処理する方式として 1990 年代まで活躍した．その後，検査項目数が増え，その依頼にばらつきが出てきて，これらの変化に対応すべくバッファ及び追い越し機能を付加した方式(STS: Side Track System)を基本方式として考案・提案した．この方式を採用したシステムは現在も稼働している．

STS では，搬送路上とは別の分析装置内バッファに検体を引き込み，そこからピペット動作が行われる．よって，バッファに引き込まれる限り他の分析装置に影響を与えない．追い越し機能では，検体は途中の不要な分析装置があれば，これを追い越して運ばれる．このバッファ及び追い越し機能により STS は PLS に比べ優位なシステム性能が期待できる．STS は 1990 年代に開発され異種の分析装置 4 台構成まで組合せを可能にしたが，そのポテンシャルは高い．開発時は製品仕様内においてシステム処理性能を検討しながら開発が行われたが，STS 方式のポテンシャルを客観的数値的に表現してその性能の可能性を追求することは今後の設計開発にとって重要である．そこで，本研究では STS のシステム処理性能を解析し，それを数値実験により裏付け，STS の優位性を引き出し，また，顧客の環境に依存しない形で搬送システムをモデル化して評価する手法を確立する．

システム処理性能の解析では，ランダムな検体列において特定装置に立ち寄らない検体が連続して出現することに着目し，立ち寄らないことで発生する空きを圧縮するアーキテクチャとして STS を位置づけた．数値実験で検証した結果，1)STS では搬送路の 1 検体当りの搬送時間が(検体の分析処理時間)/(装置数+1)を超えると処理時間性能が大きく劣化する，2) 引き込みラインの入口出口それぞれにバッファを 1 個もつ装置 4 台構成において，ランダムな分析装置間経路をもつ全 200 検体を検査処理終了するまでの時間は，STS が PLS に対して平均 30%圧縮する，3)STS のバッファ 2 個以上にしても更に 30%圧縮するほどではないことが明らかになった．これにより STS

の優位性を定量的に確認し，プラットフォームにおける搬送路及びバッファの設計基準を構築した．

（2）プラットフォーム開発手法

プラットフォームを実現する開発プロセス手法は，前述した設計知見をコア資産となるソフトウェアで実現し，広く複数の製品において共用する手法である．ソフトウェアのコア資産化手法としては，組込みソフトウェア開発手法として注目されているSPL(Software Product Lines)手法がある．本研究ではこの SPL の考え方を導入し，異種の分析装置を統合するプラットフォームソフトウェア(Analyzer Integration Management System, 以下 AIMS と略す)の新しい開発手法を構築している．AIMS に新たな分析装置を接続するためには，分析装置に特有な管理が必要となるため，AIMS 内に広範囲な改造が発生し開発計画が難しくなる．加えて，複数の分析装置の開発と連動して AIMS が開発されるため，開発サイクルは比較的長期であり，コア資産を抽出するために参照する過去のソースコードも限られている．

そこで少ない参照ソースコードでも精度よくコア資産を抽出・見積りする手法として，AIMS のアーキテクチャ要素を更に検査室の業務フロー要素単位に分解して，コア資産とアプリケーションを区分け解析する Architecture Domain Matrix (ADM)手法を考案した．この手法により見積り精度が高まるだけでなく，改造部位を業務フロー要素毎にまとめると Work Breakdown Structure (WBS)が作成でき，アーキテクチャ要素毎に集計すると改造量に見合った開発チーム編成に役立つので開発プロセスの生産性向上が期待できる．本開発手法を実プロジェクトに適用したところ，組込みソフトウェアにおいて3機種接続を1.5年で完了させることができ過去の開発に比べ2.5倍の生産性を実現することができた．また，ここで抽出されたコア資産は，その後 4 年間で新分析装置 5 機種，新搬送路 2 機種接続に活用されており，コア資産として活用されている．これは抽出されたコア資産が実効的であったことを示す．

以上(1)(2)の成果は自動分析装置の開発に適用された. 搬送制御技術により異種分析装置を搬送路で結合した業界初のシステム製品が構築され世界市場で高いシェアを維持している．また，開発手法を適用することにより高い生産性を達成でき，これもシステム製品の開発促進に大いに寄与した．

以下各章の概略を説明して，本論文の構成を示す．第 1 章では研究の背景と目的を

述べる．第 2 章では本論文の問題設定を行う．検体検査プラットフォームにおける問題として，搬送制御の性能評価，及び，プラットフォーム開発プロセスを捉える．

第 3 章では，STS とその搬送制御実現方式を示し，検体の装置経路ばらつきが大きい場合に出てくる旧来方式の処理性能低下を STS が克服することを示す.

第 4 章では，STS と旧来方式の処理性能を解析的に比較する．まず，STS が性能を維持するために必要な限界搬送能力を解析する．次に，性能比較に用いる新しい指標として検査列に発生する空きの確率分布を定義・導入し，搬送システムが同確率分布を変換するフィルタにモデル化できることを導き，STS が優位になるメカニズムを明らかにする.

第 5 章では，第 4 章の解析結果を数値実験により検証する．STS がもつ限界搬送能力の存在を検証し，空きの確率分布のフィルタにモデル化できることを数値で示し，旧来方式に対する STS の処理時間圧縮率を算出して STS の優位性を検証する. また，STS の装置台数とバッファ数がシステム性能に与える影響も数値実験して論ずる.

第 6 章では，STS をプラットフォームに実現するソフト開発手法として，ADM(Architecture Domain Matrix)手法を提案する．分析装置を接続するたびに発生する改造が散在することを課題として，アーキテクチャ要素とドメイン要素の両面でソースコードを分類することにより見積り分解能を上げ，ADM 手法が開発コストをコントロールできることを示す.

第 7 章では，ADM 手法を実際の開発プロジェクトに適用した結果を述べ，ADM 手法による生産性向上を数値で示す.

第 8 章では，STS の搬送制御方式とそのソフト開発手法を実際に適用した製品実績について述べる．1990 年代に始まった大型自動分析装置の開発で考案・採用された STS は，顧客の環境に合わせた分析装置の組合せを可能にしただけでなく，検体の追い越しにより検体の渋滞を防止することができるという特長を生み出した．これにより世界市場における高いシェアを築くことができた．後に大型のみならず中型自動分析装置の開発にもこの方式は引き継がれ，更に，生産性向上のためのプラットフォームが前述の ADM 手法により開発され，これらをベースに多機種に渡る自動分析装置システムを高い生産性で構築し，自動分析装置の世界市場における高いシェアを維持

している．

第 9 章では, 以上述べた STS の搬送制御方式及びソフト開発手法における課題に対してどのような解決がなされたかをまとめ，それらがシステム製品に貢献したことを述べる．

以上の研究の成果として，設計から実装に至る開発全体において搬送プラットフォーム構築手法を確立して性能向上及び開発効率向上を実現でき，自動搬送制御を組み入れることができた．この手法を採用した検体検査システムはワールドワイドに利用されており，検査コストの低減及び検査の迅速化に貢献している．

目次

第１章　序論

1.1 節では検体検査において自動搬送が必要とされた背景を述べる．

1.1.1 節は検体検査の自動化が果たしてきた役割を説明し，1.1.2 節は検体搬送の自動化において検体という物と検査情報の流れを説明して，自動化の対象を示す．1.1.3 節は搬送の自動化に関する歴史を紐解き，特に 1990 年代に新たに開発された方式について述べ，その方式がどの点で優位かを述べる．1.1.4 節では自動搬送をプラットフォーム化する上で焦点を当てるべき開発プロセス中の工程を述べる．

1.2 節では，以上の背景を基に研究の目的を設定し，1.3 節では，本論文の構成を説明する．

１．１　研究の背景

１．１．１ 検体検査

医療においては診療を支援するために，例えば，X 線によるレントゲン撮像や血液検査といった検査が行われる．本論文が取り扱う分野は検体検査と呼び，これは人から採取された血液や尿などを検査対象とする分野である．

検体検査の自動化のために検査室では分析装置が使われている．中でも生化学的検査と呼ばれる分野の分析装置は分析処理性能が著しく向上してきた．生化学的検査と

は人間ドックで典型的に検査される肝機能やコレステロールといった成分の検査である．この測定を行う分析装置では，この数十年の間技術が進歩し，検査可能項目数，処理スピード，測定精度が大幅に向上し，装置を動作させるための手間が減少しランニングコストが低減された．検査項目数については 1980 年以前の装置が 10 項目程度だったのに対して現在では 150 項目を可能としている[1]．その結果，自動化を通じて検査コストの抑制には貢献してきたといえる．

自動化が必要とされた背景はこのようなコスト低減だけではなく，迅速検査のニーズも高い．病院において患者に迅速な医療サービスを提供できる体制は重要である．その意味で分析装置の処理速度向上に対するニーズは高い．

このような分析装置は病院内の検査室に集中的に配置されている．入院または外来の患者から採取された血液は一括して検査室に運ばれ検査される．検査結果は印刷物または電子媒体にて医師に報告される．

検査室を設けない医療機関は，検査を専門に行う業者である検査センターに検体を送付する．検査センターは大量な検体を高速に処理するため，検体検査の自動化が必須となる．検査センターに配送された検体の検査結果は，依頼をした医師に報告される．

日本のみならず諸外国においても検体検査の自動化は進んでおり，発展途上国では更なる進展が想定される．経産省[2]によれば血液検査装置は輸出が際立った医療機器であり，医療市場の伸びも想定されている．

以上をまとめると検体検査の自動化は，

- ランニングコストの低減により検査コストを低減
- 迅速検査に対するニーズ
- 世界的な医療市場の需要

を背景に求められている．本論文は，分析装置を搬送路に結合して組み合わせることによる自動化を対象としている．これにより，別々に配置されていた分析装置に対して人が検体を運ぶという業務フローが自動化され，更なる検査コスト低減，検査迅速化が可能となる．

1.1.2 検体検査の業務フロー

本論文における自動搬送機能は，検体検査の業務フローを自動化しているともいえる．そこで，病院内検査室を例に，分析装置に検体が処理される前後を含めて，検体という物と検体情報がどのような関連性をもって管理されるかを概観する．生化学的検査を中心に図 1‑1 を参照しながら説明する．

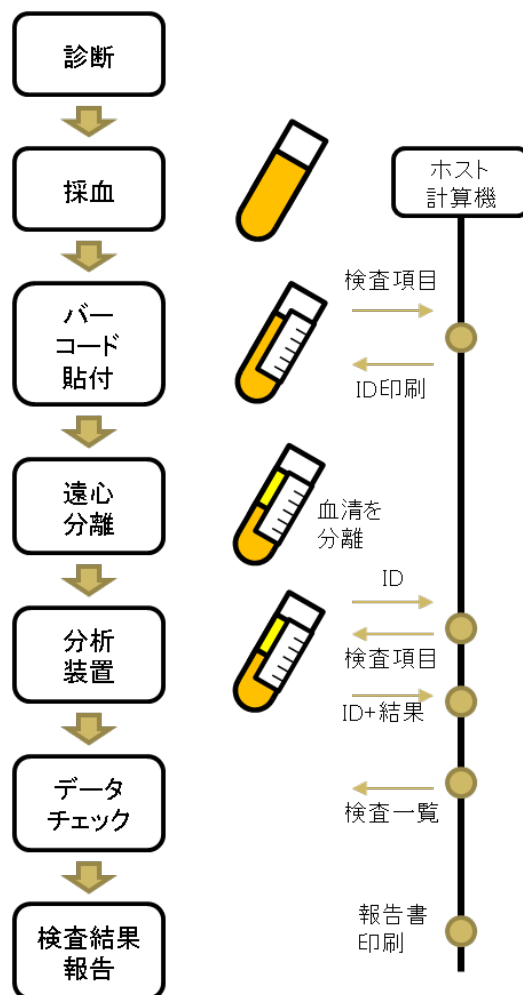



図 1‑1　検体検査の業務フロー

医師による診断が行われ,診断に必要な血液の検査が依頼されると採血が行われる．血液は採血管に封入され，採血管の検査依頼項目がホスト計算機に入力される．ここでホスト計算機とは主に医事会計を行う計算機であり，検査の課金もここで処理され

る．検査依頼項目の入力と同時にバーコード(図中 ID)が発行されるので，これを採血管に貼付する．このバーコードが後に検体情報を照会するために役立つ．

全ての検体は病院の検査室に運ばれる．検体の生化学的検査を行うためには遠心分離処理が必要となる．生化学的検査においては血液を遠心分離した上澄みである血清の成分分析を行うためである．一般的に処理には数十検体一括して 10〜20 分を要する．遠心分離された検体は順次分析装置に設置され分析が行われる．

分析装置では検体に貼られたバーコードを読み込みホスト計算機に依頼検査項目の問い合わせをする．ホスト計算機ではバーコード発行時点に関連付けられた依頼検査項目を分析装置に指示する．分析装置ではこの依頼検査項目に従って採血管から血清を一定量吸い取って装置内反応容器に吐出する．このように液を小分けする動作を分注と呼ぶ．反応容器では依頼検査項目に応じた反応と透過する光の変化が計測され濃度が算出される．分析装置はこの値をバーコードの ID とともにホスト計算機に送付する．

一般に依頼検査項目毎に反応容器への分注が発生するため，依頼検査項目数に分注周期時間を乗じた時間だけ検体は分析装置によって束縛される．

検査結果が揃った段階で検査技師によりデータチェックが行われる．過去の検査結果や精度管理情報などを参照して，検査結果が正しいかが検証される．検査の信頼度をより高めるために検体によっては再度検査を行う場合もある．これを再検と呼ぶ．データチェックが終わると結果が医師に報告される．

以上が検体と情報の流れである．緊急に検査する必要のある検体を緊急検体と呼ぶが，これも同じフローを通る．一般検体との違いは，分析装置が優先的に検査を行うことが検出できるように特定のバーコードを発行する，または，持ち回りで検査するなど，優先処理の工夫が施される．

本論文の自動搬送機能は，搬送路で結合された複数の分析装置を前提としている．搬送路で結合されていない場合には，検査を必要とする分析装置を巡って人が検体を持ち回るか，分析装置毎に検体を別容器に小分けするなどの手間がかかる．または，採血時点で分析装置別に採血管が用意される．分析装置毎に割り当てられた検体は，

それぞれ対応する分析装置に設置され分析が行われる．

このような複数の分析装置で分析されることを想定した検体が本論文の自動搬送機能によって処理される．検体が搬送路入口に到着すると，検体のバーコードは搬送路入口で読み取られる．ホスト計算機に問い合わせることによりに検査項目が分かる．自動搬送機能はあらかじめ各分析装置が分析できる検査項目群を登録している．この登録データから，検体が搬送すべき分析装置経路が算出できる．自動搬送機能は検体の分析装置経路に基づいて検体を分析装置に搬送しては分注を行うという動作を繰り返し，全ての搬送・分注が終わると搬送路出口に搬出して，検体収納部に収める．

以上をまとめて，業務フローにおける省力化を行うために自動搬送機能は以下の処理を行っている．

- 検体に貼付されたバーコードにより検査項目が伝達される
- 検査項目は分析装置経路に変換される
- 変換された分析装置経路に従って搬送・分注を繰り返す
- 分析装置から出力される分析結果を検体単位にまとめて報告する

１.１.３ 自動搬送の歴史

検体検査を自動で行う自動分析装置は，生化学分析の分野で進展してきた[3]．分析装置のプロトタイプは 1950 年代に海外の会社により開発され，当初は検査項目数も限られていた．その後方式を変えながら，検査項目の数が増え，分析装置は進化してきた．自動分析装置は国内にも輸入され活用されたが，国内メーカーもその進化の中で大きな役割を果たした．中でも，反応ラインに無駄な空きをなくし，多項目を効率的に分析するランダムアクセス方式が検査の高速化に大きく貢献した．装置の改良だけではなく新しい検査項目の試薬開発とともに自動分析装置は益々その需要を高めている．

分析装置を搬送路に接続する方式は，検査室内の検査自動化(ラボラトリーオートメーション)システムの開発によって多様な接続を可能にした[4]．これは，検査室の機器構成に応じてカスタマイズしてシステムを構築している．加えて，大量高速に検査

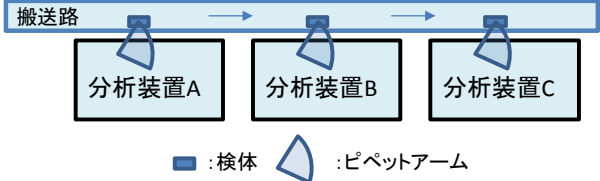
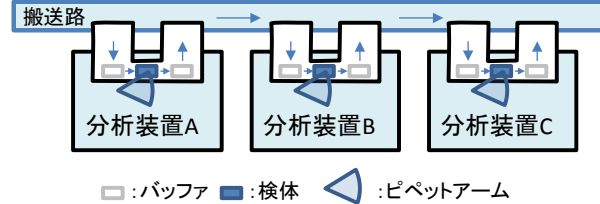
するというニーズに応えて，搬送路を装置内に内蔵して複数の内部分析装置を結合する大型分析装置も開発された．

方式名称と構成図	特徴
(a) Pipe Line System (PLS) (～1990年代) 搬送路 分析装置A 分析装置B 分析装置C :検体 :ピペットアーム	搬送路上に検体を並べて，同期シフトする方式 検体毎に検査項目数が比較的均一である場合，大量高速に処理
(b) Side Track System (STS) (1990年代～) 搬送路 分析装置A 分析装置B 分析装置C :バッファ :検体 :ピペットアーム	分析装置側に引き込みラインをもたせ，干渉なく分析を行う方式 検査項目数のばらつきがある場合，異機種を組み合わせて効率的に処理

図　１-２　大型分析装置の実現方式

図　１-２は 1990 年代を境に分化してきた大型分析装置の 2 実現方式を並べて比較している．本研究では図(a)の方式を Pipe Line System (PLS)，図(b)の方式を Side Track System (STS)と呼ぶ．PLS は，搬送路沿いに複数の分析装置を直付けして，搬送路上に流れてくる検体容器から各分析装置が検体を採取する方式である[5]．STS は，搬送路上に流れてくる検体容器を分析装置側に引き込んで搬送路とは独立に検体を採取する方式であり[18]，筆者はこの基本方式を考案し，製品に適用した．この方式については 1996 年には国内特許[6]を，2003 年以降，米国及び欧州の特許[7][8]を取得している．

大型分析装置の仕組みは 1990 年代に PLS から STS へと変わった．その変化の背景には，自動分析装置によって検査できる検査項目数が増加してきたこと，従って 1 検体に依頼される検査項目が多様になったことがあげられる．生化学的検査に加えて，人体の中で外敵から身を守る反応がどのように行われているかを測定する免疫学的検査についても分析装置が発展してきた．このような複数の分析装置を組み合わせて，様々な検査依頼をもつ検体を効率的に検査する需要が高まってきた．

PLSでは，比較的検査項目数が揃っている検体を流す場合に効率のよい検査の自動化が行われる．このためPLSは同種の分析装置を同期して接続する形で1990年代まで，分析装置を搬送路で結合してシステム全体の処理性能を向上してきた．一方，検査項目が多様な検体を流す場合には，STS のような引き込みラインが有効に作用する．異種の分析装置4台構成まで組合せを可能にした製品にてSTSは採用され,現在も稼働している．製品としてその処理性能は検証されてきたが，方式としてSTSが有するポテンシャルは高く，今後の設計開発にとってその性能の可能性を追求することは重要である．

そこで，本論文は，STSにおける搬送制御システムの基本方式を提案し搬送能力などのパラメータがシステム処理性能に与える影響を明らかにした上で，STSがばらつきある検査項目に対処する様子を定量的に明らかにする．これによりSTSの優位性が客観的に明らかになるだけでなく，搬送システムの1評価方法も確立する．

1.1.4 プラットフォーム

検査効率を向上させる自動搬送機能が選択された場合，その仕組みをできるだけ多くの製品で活用する仕掛けが必要である．一般社団法人 電子情報技術産業協会の報告[9]によれば，日本の組込みソフトウェアの開発規模は大規模となって，調査したプロジェクトの48%が1000kLOC(Line Of Codes)を越えるプログラムを開発している．加えて，92%が複数機種を並行開発している．従って，組込みソフトの対象業務における知見が共用できるようなプラットフォームの開発手法が必要になる．このようなプラットフォームをコア資産としてたくさんの機種で共用することができれば，開発量を抑制しながらより多くの顧客満足を得ることができる．

本研究の対象である検体検査システムも前述の 48%に含まれるほどに大規模であり，自動搬送機能をプラットフォーム化する必要がある．システム性能が優れる制御方式が決定されても，それが多くの分析装置接続に活用されなければ意味が無い．また，一定の標準的な接続方法を規定して，顧客施設のニーズにマッチしてフレキシブルに多種の分析装置が接続できるようなプラットフォームが求められる．システム性能設計と同様にプラットフォーム化技術はハード，ソフト両面で重要となる．

本研究ではソフトウェアプラットフォーム化の中でも開発見積りの工程に焦点を当てる．搬送システム全体の開発は大規模であるため，コア資産(新たな分析装置を接続しても変わらない部位)とそのアプリケーション(新たな分析装置を接続するたびに変更する部位)をゼロから開発することは困難である．既存ソフトウェアを解析してコア資産を抽出して新たな複数の分析装置を接続開発する方が現実的である．このような既存ソフトウェアを改造する場合，見積り工程は重要である．後に見積り時の方針がぶれれば，必要以上のコードを作りすぎたり，インタフェースにおいて思いがけないコードを作らざるを得なかったり，開発期限の維持が困難になる．見積り工程は，改造方針を設計する工程といってもよく，この工程でどのようなコア資産をプラットフォームに埋め込むかを方針決定して，開発ではその方針を守り，結果的に見積り精度を向上させることが肝要である．本研究では見積り工程に焦点を絞る．

以上をプラットフォームについてまとめれば，以下となる．

■ より多くの分析装置をフレキシブルに接続できるプラットフォームが望まれる
■ プラットフォーム化は開発量抑制の点でも重要である
■ 何をコア資産としてプラットフォームに実装するかを見積り工程で決定する
■ 本研究では見積り工程における開発プロセスに焦点を当てる

1.2 研究の目的

本論文は検体検査における自動搬送制御及びその統合システムを実現する手法について報告する”システム開発型論文”である．本研究は，検査業務ワークフローを省力化できる自動搬送プラットフォームにより検査ランニングコストを低減し検査迅速化を目指す．

具体的には，

(1) 処理性能を向上させる自動搬送制御の設計

(2) フレキシブルに分析装置を接続できるプラットフォームの開発手法

の確立を目的とする．

自動搬送プラットフォームは，検査室負荷に対応してフレキシブルに装置を構成でき，かつ，検査効率を向上させる．分析装置単体であれば，個別に検査効率向上に向けて性能向上を目指さなくてはならない．しかし，分析装置の組合せにより更なる処理性能向上を目指せるとすれば，従来の分析装置に埋め込まれた性能を活かしながら，更に価値の高いシステムを検査室に提供することができ，品質，コストともに良好なシステムが可能となる．

また，そのようなプラットフォームを開発する技術では，異なる分析装置に対する多様な情報管理と搬送制御管理とが交錯する複雑性を見積り工程において整理して，コア資産として埋め込む機能を決定する．見積りにおいては分析装置単体がもつ特異性を維持しながら一方でプラットフォームとしての標準的な枠組みを用意する必要がある．コア資産と製品の開発が始まる前に見積り工程で特異性と標準の方針を明確にして，ソースコードの改造計画にまで落とし込み，開発に備えることが全体開発にとって重要となる．

１.３　本論文の構成

本論文では，1990 年代から現在まで報告者が行った，複数分析装置を自動搬送路で接続したシステムのソフトウェアプラットフォームの検討と，このようなソフトウェアプラットフォームの開発手法の検討，並びに，この実システム適用における評価に関して報告する．

以下各章の概略を説明して，本論文の構成を示す．

第 1 章ではこれまで述べてきたように，検体検査自動化における迅速化ニーズと製品化の歴史やプラットフォーム開発における見積りの意義を述べ，検査委検査プラットフォームにおける搬送制御設計及び開発手法の大きく 2 点を研究の目的とすることを述べた．

第 2 章では本論文における更に詳細な問題設定を行う．検体検査プラットフォーム

における問題として，搬送制御のモデリングと性能評価，及び，コア資産を意識したプラットフォーム開発手法を捉える．

第 3 章では，搬送制御プラットフォームの実現方式であるサイドトラック(STS: Side Track System)方式の搬送制御方式を提言する．STS は旧来のシステム実現方式であるパイプライン(PLS: Pipe Line System)方式に比べて検体の装置経路にばらつきが有る場合に優位であり，この優位性を引き出すための搬送制御方式を提案する．

第 4 章では，PLS と STS を処理性能比較し，検体の装置経路にばらつきが有る場合に STS が優位であることを解析的に導出する．まず，STS が性能を維持するために最低限必要となる限界搬送能力を導出する．次に，性能比較に用いる新しい指標，検査距離確率分布を定義・導入し，検体列を投入してから処理し終わるまでの時間(makespan)が短縮されるメカニズムをモデル化する．モデル化では，搬送システムを入力となる検査距離確率分布から出力となる検査距離確率分布に変換するフィルタに位置付ける．

第 5 章では，第 4 章の解析結果を検証するために数値実験を行い，その結果を考察する．STS がもつ限界搬送能力の存在を検証し，PLS に対する STS の makespan 圧縮率を算出して検査距離確率分布のフィルタでモデル化できること及び具体的な圧縮率を算出して STS の優位性を検証する．また，STS が製品実現されたときの仕様である装置台数及びバッファ数を超えた場合の処理性能についても実験して論ずる．

第 6 章では，第 4 章，第 5 章で優位性が確認された STS をプラットフォームに実現する手法として，ADM(Architecture Domain Matrix)手法を提案する．分析装置を接続するたびに発生する改造が散在することを課題として，アーキテクチャ要素とドメイン要素の両面でソースコードを分類することにより見積り分解能を上げ，ADM 手法が開発コストをコントロールすることを示す．

第 7 章では，ADM 手法を実際の開発プロジェクトに適用した結果を述べ，ADM 手法の有効性を述べる．ここでは ADM 手法による生産性向上を数値で示すとともに，ADM 手法が特有にもつ利点を考察する．

第 8 章では，以上述べた搬送制御方式，及び ADM 手法を実際に適用した製品につ

いて述べる．1990 年代には大型自動分析装置における搬送方式として STS は採用され業界初の方式としてアピールされた．その後結合される分析装置が増え世界市場において大型自動分析装置の高いシェアを築くことができた．大型のみならず中型自動分析装置にも方式は導入され，また，プラットフォーム構築に本論文で論じる ADM 方式が使われ高い機種生産性を得ることができた．これらにより開発された自動分析装置は世界市場で高いシェアを維持している．

第 9 章では, 以上述べた STS の搬送制御方式及びソフト開発手法における課題に対してどのような解決がなされたかをまとめ，それらがシステム製品に貢献したことを述べる．

第２章　問題設定

本章では，第１章で述べた研究の目的に従って，本論文で扱う搬送制御及びプラットフォーム開発に要求される事項を明らかにして，研究の課題を設定し，方針を指し示す.

2.1 節では，搬送制御方式の要件として処理性能上の課題を明らかにする．2.1.1 節では搬送制御システムをオンラインスケジューリングと位置付け, 2.1.2 節では旧来方式である PLS との比較において STS の処理性能が優位であることを示すための課題設定を行う. 2.1.3 節では関連する研究として待ち行列理論やスケジューリング理論や他文献との比較を行う.

2.2 節では，搬送制御プラットフォームにおけるコア資産構築上の課題を明らかにする. 2.2.1 節では, 分析装置を容易に接続できるプラットフォームが必要となることを示す．2.2.2 節では，プラットフォームを AIMS と名付け複数の分析装置を組み合わせた１つの大きな分析装置と位置付ける. 2.2.3 節では分析装置を接続することにより AIMS のアーキテクチャ要素全てに変更がおよび準備なしの開発は困難であることを示す．2.2.4 節は困難を解決する技法である SPL(Software Product Line)手法を導

入し加えて非量産系の製品特有の課題解決が必要となることを述べ, 課題設定を行う. 2.2.5 節では関連する研究として, SPL, フィーチャモデル, 他の文献との比較を行う.

2.3 節では，2.1 節及び 2.2 節の課題設定を受けて，3 章以降の章立てにおいて章間がどのように関連して課題解決を構成しているかを述べる.

２.１　搬送制御の設計

２.１.１ オンラインスケジューリング

搬送路による自動化が無い場合，一般に，人が検体を分析装置に運び分析装置内に設置して分析が行われる．複数の分析装置に跨る検査依頼がある場合には，次の分析装置に人が持ち運ぶ方法か，または，あらかじめ検体を分析装置に対応した別容器に小分けする方法がある．後者では採血管を親検体と呼び，そこから検体を小分けした試験管を子検体と呼ぶ．子検体にも親検体と同様にバーコードが貼られている．子検体を集めて，対応する装置に設置すれば装置間で検体を持ち運ぶ必要なく検体検査が行われる.

図　2-1 は複数の分析装置に跨る検査依頼がある検体の例である．検体 1 は装置 1 と装置 3 で検査され，検体 2 は装置 2，3，検体 3 は装置 1，2 で検査される．上記した小分けは例えば親検体 1 からは子検体 1-1, 3-1 へ行われる．子検体は装置毎に集められ，装置内に設置され分析が行われる.

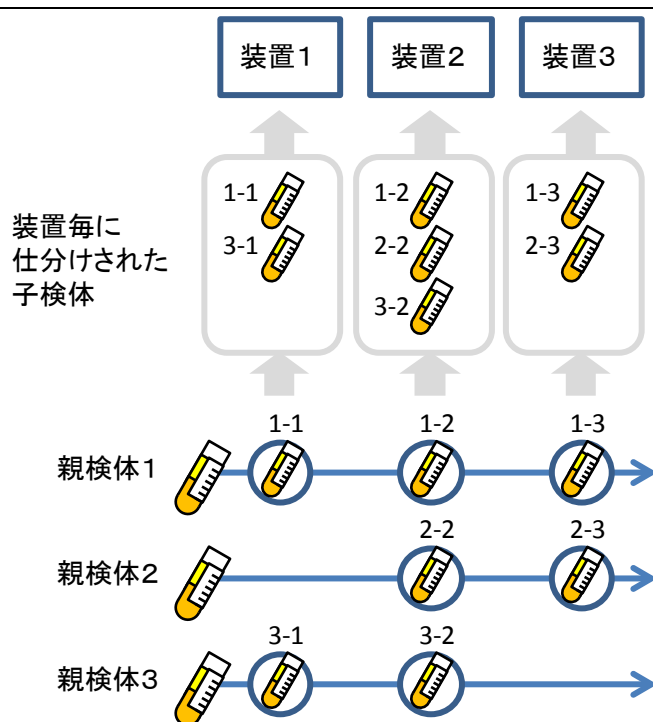



図　2-1　自動搬送が無い場合の検体の流れ

一方，自動搬送付きのシステムを使うと以上のフローが簡素化され，小分けを必要とせずに親検体を直接システムに投入することができる．例えば，親検体 1 は装置 1 での検体採取が終わると自動搬送により装置 2 に運ばれて，装置 2 の検体採取が行われ，装置 3 についても同様に順次検体採取が行われる．

このような搬送システムでは，検体が検査室に到着した順に搬送可能な容器に設置され，まとめて順次投入される．搬送路入口において検体に添付されたバーコードを読み込んで初めてその検体が必要とする検査が分かり，どの分析装置に搬送されるべきかが分かる(図　2-2)．このように投入される全ての検体の行き先は事前に分かっているのではなく，検体が投入される毎に順次行き先が分かる．検体と分析装置をジョブショップ・スケジューリング問題[10]におけるジョブと機械の関係と見立てることはリソースの対応としては適切に見える．しかしジョブショップ・スケジューリング問題では全てのジョブについてどの機械で処理されるかが事前に分かっていて，これを目標関数の元で最適化する．一方検体の搬送制御では検体が投入される毎に検査内容が明らかになっていくため，ジョブショップ・スケジューリング問題にはそぐわない．よって，随時流れてくる検体を自動的に分析装置へ配送するためのオンラインス

ケジューリング問題として捉える．

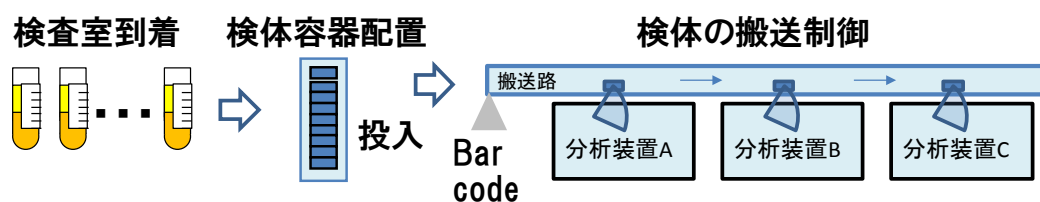



図 2‐2　検体の到着と搬送路投入

システム内では複数の検体が１つの分析装置の検体採取について競合する．このため，競合を避けて順次処理していくための制御が必要となる．そして，システム全体としてどれくらい短期に一定の検体数を処理できるかという処理性能によってその制御の善し悪しは決する．

２.１.２ 搬送制御の課題

パイプライン方式の構成を図 2‐3 に示す．分析装置は搬送路上に停止した検体容器からピペットにより検体を一定量吸い取り，分析装置内の反応容器に吐出する(以下，この一連動作を分注と呼ぶ)．検査項目毎に反応容器が用意され，分注は検査項目数分繰り返され，反応後に結果が報告される．搬送路沿いにはこのような分析装置が複数個並び，搬送路上に複数個並んだ検体の内分注時間が一番長い検体を待って，搬送路がシフトする．検体はシフトによって順次異なる分析装置の前に送られ分注が行われる．

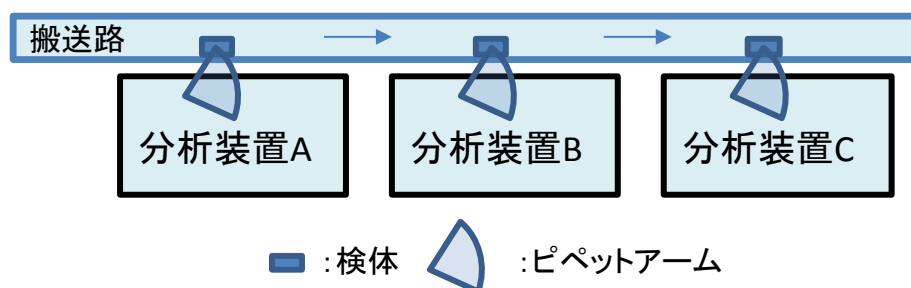



図 2‐3　パイプライン方式

このようなシステムにおいて分析装置間で分注時間に大きな差が生じると一番時間のかかる分析装置がボトルネックとなる．ボトルネックの分析装置が分注している間他の装置は分注が行えないため稼働率を低下させ，期待されるシステム処理時間性能(一定の検体数を全て処理完了するまでの時間. 以下断りが無い限りシステム性能と書けばこれを意味する)を引き出せない可能性がある．このため，システムを導入する前に，分注時間が均一になるように分析装置の種類や分析装置の検査項目レパートリーを最適化設計するのが一般的である．このような最適化では検査室での平均的な検査項目の依頼状況を参考にする．

PLS は同種の分析装置を同期して接続する形で 1990 年代まで，分析装置を搬送路で結合してシステム全体の処理性能を向上してきた．しかし，検査項目の種類も増えて，複数の分析装置で処理するとき経由する装置がばらつくような検体に対処して，1990 年中ごろに新しい接続方式であるサイドトラック方式(Side Track System: STS)が開発され，今日現在も製品に実装されている．検査項目の依頼にばらつきがあると各装置で分注時間は不均一となる．従って，PLS では搬送ライン上で分注が早めに終わる検体が停滞して，対応する分析装置の稼働率が低下する．

STS はこのような問題に対処するため，ハード構成としては分析装置側に引き込みラインを設けて，搬送路とは独立に分注が行われる．一方，制御的には次の 2 点が改良点となる．１つはバッファ機能である．分析装置は搬送路上とは別のバッファに検体を引き込み，そこから分注が行われる．よって，バッファ経由で分注が行われる限り他の分析装置に性能上の影響を与えない．もう１つは追い越し機能である．検体はある分析装置で分注を終えると，後続の任意の分析装置のバッファに運ばれる．途中の不要な分析装置があれば，これを追い越して運ばれる．

STS は異種の分析装置を 2 台から 4 台まで組合せ分析装置内には分注前と後に１つずつバッファを配置して製品として実現されたが，そのポテンシャルは高く，今後の設計開発にとってその性能の可能性を追求することは重要である．

そこで，本論文は，バッファ数及び搬送能力がシステム処理性能に与える影響を明らかにして，STS がばらつきある検査項目に対処する様子を定量的に明らかにすることを目的とする．

具体的には，STS において処理性能を向上させる搬送制御方式を導出した上で，

課題１：STS のシステム処理性能を引き出すための前提条件となる最低限必要な搬送能力(限界搬送能力と呼ぶ)

課題２：STS が検体のばらつきに対して処理性能を向上させているメカニズムと旧来方式 PLS に対する優位性

課題３：分析装置８台，バッファ数３つまでの製品仕様を超えた構成における STS の処理性能

を明らかにする(図　2-4)．特に課題　2　の解析では，ランダムな検体列における空きの分布を導出し，それが搬送システムによって縮小されるモデルを導出する．

STS製品仕様
分析装置台数2～4
バッファ数1
解析
(a) 処理能力の前提条件：
搬送路の限界搬送能力
(b) 検体のばらつきに対応した処理能力向上のメカニズム

数値実験
(a)(b) 上記解析の検証
(c) 仕様拡張時の処理性能
分析装置台数2～8
バッファ数1～3

STSのポテンシャルを引き出し
今後の設計に役立てる

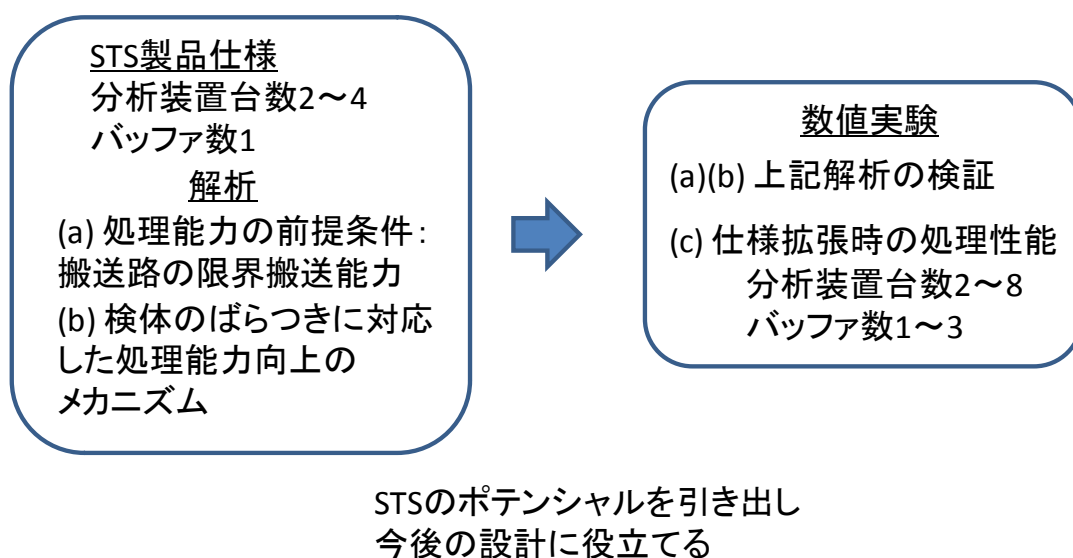



図　2-4　STS の性能解析と検証

２.１.３ 性能評価に関連する研究との比較

(1) 待ち行列理論

物を自動搬送する設備は生産現場においてよく見られる．生産性を向上させる手段としては基本的な設備となる．複数の加工機械の間を自動搬送で結合するシステムでは，製品が機械に搬送され一定時間をかけて加工処理が施され，その後加工された製品が更に次の機械に搬送される．

このような機械と搬送を組み合わせたシステムの性能を分析するツールとして待ち行列理論[11]が知られている．待ち行列理論では機械の前に並ぶ待ち行列の平均長や

平均待ち時間を算出することができ，これによりシステムとして用意すべき待ちスペースを設計することができる．

検体検査システムでは前述したように検査室に到着した検体はまとめてシステムに投入されるが，各装置に関して言えば検体がある確率分布で到着することから，待ち行列理論を適用する可能性がある．しかし，分析装置の前後にあるバッファは有限数のため，もしバッファが一杯であれば待ちが上流に連鎖する現象，すなわちブロッキングが発生する．バッファが一杯になると検体の搬送元である上流の分析装置において処理が終了した検体を次に渡すことができなくなり遂にはこの上流の分析装置までも動作できなくなる．これをブロッキングと呼ぶ．

ブロッキングを考慮した直列型待ち行列については 2 機械であればその待ち行列を容易に計算することができる[12]．これによれば，待ち行列長が n となる状態だけでなくブロックされる状態を含めて確率を計算する．しかし，機械が増えるに従って状態は複雑となり一般的な計算式の導出は難しくなる．従って，本論文の搬送システムでは待ち行列理論の適用が難しい．

(2) スケジューリング理論

スケジューリングとは，多くのジョブ(仕事)を乏しい資源を用いて効率よく，時間内に実行しなければならないとき，必要な資源をいつ，どのジョブにどれだけ割り当てればよいかを決定する計画である[13]．これによれば複数の分析装置(資源)に検体(ジョブ)を割り当てるスケジューリング問題として定式化は一応可能となる．

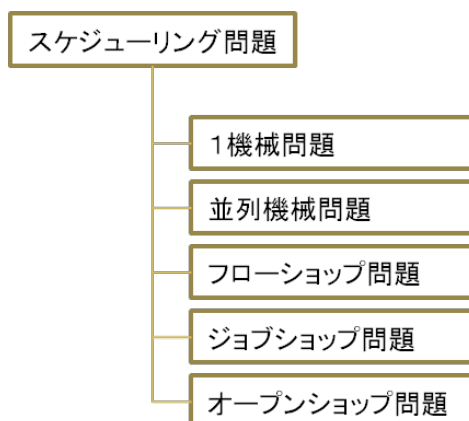



図 2-5　スケジューリング問題の分類

スケジューリング問題(Job Scheduling Problem，以下 JSP)は図　２-５に示すように制約を変えたいくつかの代表的なモデルとして研究されている[13]．中でもジョブを機械に割り当てる問題としてジョブショップ問題が論じられている．本研究における検体をジョブに，分析装置を機械に対応させるとジョブショップ問題と定式化できる．しかし，前述したように検体は到着して初めてその依頼された検査項目が分かる．このため事前に全ジョブの機械割当計画が分かった上で割当て計画を最適化するジョブショップ問題にはそぐわない．また，投入された検体を一時的にプールして順序を入れ替えるというような装置を追加してスケジューリング問題に近くすることも考えられるが，一般の検査室には分析装置以外のスペースを割くことは難しく，コスト面でも難しい．これらを人手で行うことは随時到着する検体を一まとめになるまで待つ時間や，何百検体を入れ替える作業時間や人件費を考えると更にコスト面で見合わない．

一方，事前に全ジョブの機械割当計画が分からないことを制約にしたオンラインスケジューリング問題[52]が研究されている．ジョブ毎にシステムに投入される時刻を属性付加し，ジョブはその時刻より前には投入できないことを条件としている．前述した通り，この問題は本論文の問題定式化に適用できる．しかし，研究[52]では，ある時刻までジョブを溜めてその中から最適なジョブを選択するアルゴリズムを対象としている．従って，やはり検体を一定数プールすることになり，研究の適用はそぐわない．

(3) その他の関連する研究

一般的な搬送システムについて，生産設備において複数のコンベアベルトを組み合わせて実現する例が報告されている[10]．これによると，搬送路に隣接する加工分析装置に製品を運ぶにあたり，複数の経路計画を同時に実現するために，複数のコンベアベルトの動作タイミングを決定している．整数計画法を用いて動作タイミングを計算しているため，流れる全製品の経路が事前に分かっている必要があり，オンラインスケジューリングには適さない．

入力系列におけるばらつきが与えるシステムの影響については，需要のばらつきが

直列型待ち行列の生産システムに与える影響を文献[14]が報告している．ばらつきは変動係数を使って表現しているが，本研究で扱うような，空きの分布の解析をしていない．

バッファに関しては，文献[15]がジョブショップ型生産システムの各工程においてスループットの最大化を目的関数にして，与えられた総バッファ量を各工程に割り付ける割合を決定している．工程に割り付けられるバッファ容量は工程の処理性能，流れる部品固有の経路に依存して決定されている．本研究では，検体がどの分析装置で分析されるかはランダムであるため，ランダム検体列の内，分析装置をどの程度スキップするかという空き情報を基に，分析装置内のバッファ数がシステム性能に与える影響を解析している．文献のようにバッファ容量を個別に動的に設定するのではなく，ランダム検体列に対応できる標準的な装置内固定バッファ数をシステム性能との関係から論じている．

搬送能力が処理性能に与える影響に関しては，文献[16]が複数の搬送機器をルールに基づき選択してジョブショップ型生産システムの総所要時間を最小化する手法を論じている．本研究の主題となる STS は，搬送手段が 1 つであり，全検体がこれを共有する点で異なる．

この他，搬送システムのオンラインスケジューリングにおいてバッファと搬送能力を検討する研究は筆者が知る限り無い．また，試行列内で連続して出現する要素列(連と呼ぶ)の確率分布について文献[17]が報告している．与えられた試行列から連の確率分布を効率的に算出する方法が論じられている．本研究は，ランダムな装置を選択する検体を対象にしており，その出現分布がモデル化されることを前提としているため，個別の計算は不要となる点で異なる．

2.2　　プラットフォーム開発手法

2.2.1 プラットフォームの必要性

分析装置を搬送路に結合する場合，より簡易な形で接続することが求められる．ここで「簡易な形」とは①異種の分析装置がもつ特殊性からは独立して接続できること，②必要最小限の標準インタフェースで検体を出し入れする仕組み，である．

このような形は計算機ネットワーク構成に理想を見いだせる(図　2-6).計算機がバス接続されたネットワーク構成は今では一般的になっている．このネットワーク構成ではバス及びその接続方法が規格化されており，どのような異種計算機，更に，装置でさえもがその処理性能の差をもちながらネットワークに接続することができる．本研究で求められる搬送システムは，ネットワーク構成における情報の搬送を検体の搬送に置き変えるくらい同様の構成が求められる．

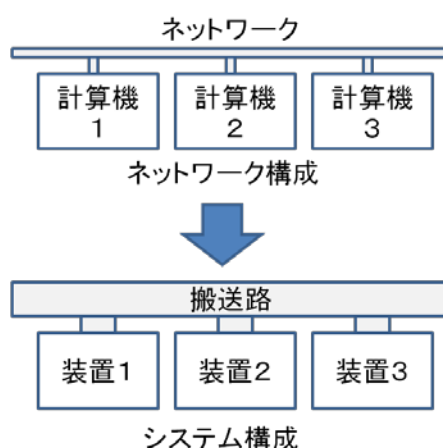



図　2-6　ネットワーク構成と相似するアーキテクチャ

システム側からみると，分析装置が組込み容易なモジュール性を有することが必要となる．モジュール性とは内部の構造を隠しながら必要最小限のインタフェースでシステム接続できる特性である．このモジュール性を実現するためにハードウェア構成を検討し，モジュール組合せ方式と呼ぶハード構成が生まれた．

モジュール組合せ方式では，モジュール性を確保するために分析装置内にローカルな搬送ラインを設けている．このローカルな搬送ラインに検体を引き込んで分析装置の処理性能に応じた速度で検体を進め，分注が終わった検体を排出することにより，反応容器の空きを最小にして実効処理性能の低下を防いでいる．また分析装置間をつなぐ搬送ラインは，任意の装置間で検体を高速に搬送することができ多様な経路を実現できる．

以上のハードウェア構成は検体検査システムにおける要件を満たす可能性をもっているものの，これだけではモジュール性という要件を十分に満たすことができない．

1 つは前述した，処理性能を引き出すための搬送制御方式が必要となる．これを本論文ではサイドトラック方式と呼び，その性能解析を行う．もう 1 つは分析装置を接続することによって起きるソフトウェアの変更が最小になるような仕組みが求められる．すなわち，ソフトウェアプラットフォーム(以下，単にプラットフォーム)という変更のない基礎ソフトと分析装置毎に必要なソフトが組み合わされて製品が構成されるという仕組みが必要となる．プラットフォームは，想定範囲にある分析装置の接続に対して堅牢であり，柔軟に分析装置を接続できるため，事業を支える資産という意味でコア資産ともいえる．このコア資産が大きくて分析装置毎の変更が少ない形が理想となる．

２.２.２ AIMS (Analyzer Integration Management Software)

一つの検体を複数の分析装置で分析する必要のある施設では複数の分析装置を搬送路で結合した，カスタマイズされたシステムを導入している．検体は適切な搬送路を経由して分析装置に自動搬送される．搬送された先ではバーコードにより検査項目がホストコンピュータから取得され分析装置で分析処理が行われる．複数の分析装置から出力される検査結果は検体毎にまとめられ，検査技師によってデータの適正を確認された後，検体毎に報告書に印刷される．このように検体を自動搬送する搬送路と分析装置を結合するシステムを本論文では医用分析装置統合システムと呼ぶ.

医用分析装置統合システムにおいては全体を統合する計算機が必要となる．この計算機は，複数の分析装置と搬送路から成るシステムを 1 つの大きな分析装置として機能させる．すなわち，ミクロには検体の搬送先分析装置を決めその分析装置に検査項目を指示し分析後の検査結果を収集する一方で，マクロには検体の検査項目を入力して検査結果を出力する．また個々の分析装置が分析に用いる試薬の管理も行う．この計算機のソフトウェアを本論文では AIMS(Analyzer Integration Management Software)と呼ぶ.

AIMS は新たな分析装置を接続するたびに改造される．この改造は，新しい分析装置が開発されるたびに発生し，そのたびに AIMS は進化していく．しかも，上位互換性の要請から進化した AIMS は過去に接続した分析装置と接続できなければならない．進化に耐えうるプラットフォームが必要とされる．

２.２.３ AIMS におけるコア資産の散在

AIMS のアーキテクチャ要素は，①UI(User Interface: 画面・帳票)，②DB(データベース)，③検体管理，④装置通信，⑤分析装置，⑥外部通信の 6 つに分割できる(図２-７)．③検体管理は検体の検査項目が決定してから検査結果が全部出揃い報告書が出力されるまで系内の検体を追跡管理し必要に応じて分析装置と通信を行うコンポーネントである．この中に，システム処理性能を決する搬送制御の核が含まれている．⑥外部通信は前述したホストコンピュータと通信を行うコンポーネントである．

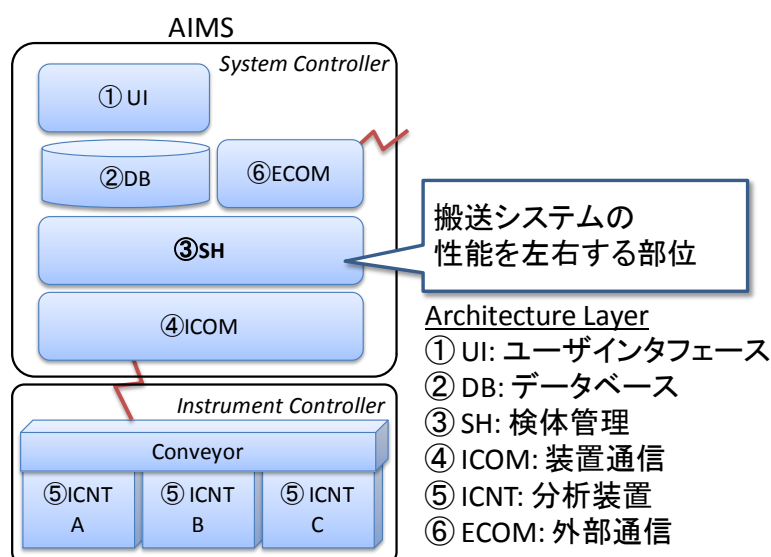



図　２-７　AIMS のアーキテクチャ

AIMS に新たな分析装置を接続するためには，①～⑥の全てを変更する必要がある．新たな分析装置と通信を行うために③検体管理，④装置通信の変更は必須である．また，分析装置が独自に持っている検査機構構成の状態をモニタするため，また，試薬を管理するために①画面・帳票，②データベースの改造が必要となる．更に，この分析装置固有の機能をホストコンピュータから利用するためには⑥外部通信の変更を余儀なくされる．

課題は，このような変更が連鎖する環境において，目前の開発計画にある分析装置の要求に合わせて全体開発コストが統制されるような見積り手段を確立することにある．

２.２.４ プラットフォームの課題

複数の分析装置を搬送路で結合するシステム化[18]を実現するにあたって，検体データと分析装置と搬送路を集中管理するシステムソフト AIMS が必要となるが，その開発には多大な開発期間と開発コストがかかる．このような組込みソフトウェアの開発が大規模になっていることは第 1 章で述べたが，別の例を挙げると，代表的な組込みソフトウェアである車載ソフトや携帯電話ソフトはプログラム規模が数百万行以上と言われる[19]．AIMS も桁数において肩を並べるほど規模が大きく開発は困難を極める．

開発量を軽減するために，装置を接続するための共通プラットフォームが求められる．このような共通プラットフォームを更新・維持していく開発手法としてはソフトウェア・プロダクトライン(Software Product Line，以下 SPL)[20][21]が知られている．この手法は，ソフトウェア開発のコア資産を管理することにより，コア資産を利用するアプリケーションの開発量を軽減し生産性を向上させる．

コア資産である共通プラットフォームを構築するために，複数の製品に分散したコードを分析する手法が報告されている[22]．実績ある既存ソフトウェア製品群を複数解析して，共通性・可変性を抽出している．AIMS においても過去に更新されたソフトウェア製品群が存在する．しかし，開発が年のオーダで長期に渡る，言わば少品種非量産系の製品であるため，共通性を見出すサンプルが少ない．従って一番最近に開発された単一の製品を基礎に共通プラットフォームを構築せざるを得ない．

また，既存ソフトウェアから共通プラットフォームを構築するにあたってはリファクタリング手法が活用できる [23]．この手法は外部仕様を変更せず再利用性や保守性を向上させるためにソースコードを改変する．しかし，この活用にあたってはどの程度改変すべきかを統制する手法を明らかにする必要がある．

本研究は計画された複数製品を開発する前に単一の既存製品のソースコードから抽出すべきコア資産を見積もる手法を述べる．見積り後は複数製品とコア資産の開発が並行し交錯するため，開発計画面で支援する．

そこで本研究は，計画された新規複数製品を，単一製品のソースコードを基礎にし

て開発するにあたり，以下 3 点の課題を解決する．

(課題 1) 見積り時点でのコア資産コスト統制

(課題 2) 全体開発量の見積り精度の向上

(課題 3) コア資産開発を含む開発計画の困難さの低減

これらに対応する施策として，まず(課題 1)については，開発予定の複数製品を可変性の範囲とみなし，複数製品間を共通化するコストと個別に開発するコストのバランスを全体計画の中で図る．

(課題 2)については，リファクタリングの解析粒度にアーキテクチャ要素とドメイン要素からなるマトリクスを導入し，ソースコードをこのマトリクス上に分類する．分類されたソースコードは機能(アーキテクチャ要素)だけでなく，その要求(ドメイン要素)からもコア資産の在り方を検討することにより，コア資産の共通性の範囲を限定する．

(課題 3)については，前述したマトリクスからコア資産開発の重みをアーキテクチャ要素単位に判定してコア資産開発者を割り当てる．複数製品が個別に実現されるにあたりコア資産開発者を含めて開発者全員で設計をレビューしテストできるツールとして，マトリクスから Work Breakdown Structure(以下，WBS)を導出する．

以上のようにマトリクスの分析を中心とした開発上流工程の強化により単一製品から SPL を導入して共通プラットフォームを構築すると同時に複数製品を開発することが可能となる．

２.２.５ コア資産開発手法に関連する研究との比較

(1) SPL (Software Product Line)

前述したように，プラットフォームの開発手法としては SPL を導入する．SPL で語られる成功事例は，例えば，ディーゼルエンジンの生産性向上，携帯電話の年当たりの開発機種数増加，プリンタの開発期間縮減などがあげられる．その多くが量産製品であり，数多くの開発経験あるいは資産から共通要素を抽出することに成功してい

る．

一方，AIMS はシステム製品であり，その開発には複数の分析装置や搬送路などシステム要素の開発が含まれている．このようなシステム製品開発の課題をまとめると以下となる．

- ■ 参考となる製品が少なく，過去の傾向からコア資産を導出することが難しい
- ■ コア資産を抽出するための参考ソースコードは最近開発された単一製品
- ■ 分析装置の開発を待って統合するため開発期間が長期に渡る

これらは，Stand-alone 製品とシステム製品の違いともいえる(図　2-8)．開発サイクルが長くなるシステム製品が抱える課題である．コア資産を抽出することが難しいために，何らかの手法により見積り時点でコア資産開発の有るべき姿を描き統制し，またその精度を上げ，また長期に渡る開発を支援する必要がある．

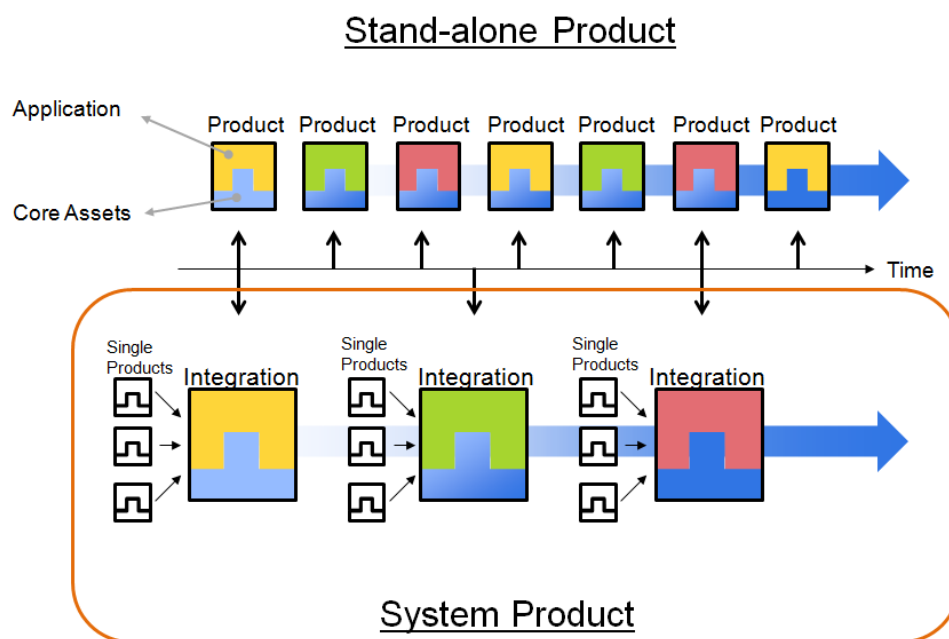



図　2-8　Stand-alone 製品とシステム製品の違い

従って，非量産製品におけるコア資産の抽出については従来の SPL に新しい施策が必要であると考える．その施策が本研究におけるプラットフォーム開発手法である．

(2) フィーチャモデル

SPL の成功事例は実践的なフレームワークとして研究され公表されている [35]．ド

メイン理解はフレームワークの一つとして重要視され，関連する手法として Feature Oriented Domain Analysis(FODA)が紹介されている[37]．これは，製品間で共通な特徴であるフィーチャ(ユーザ視点のシステム特性)を抽出し，その組合せをモデル化し，個別製品を開発するベースとしている．

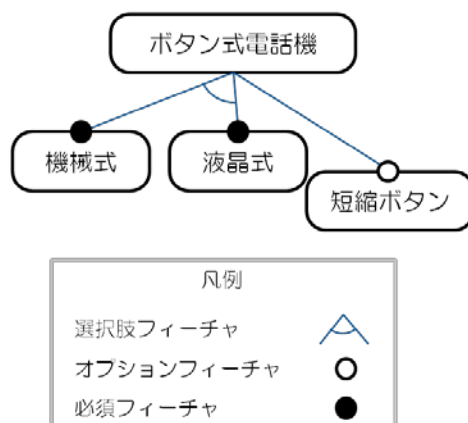



図 2-9　フィーチャモデルの例

図 2-9は簡単なフィーチャの例である．ボタン式電話は機械式か液晶式のいずれかが必須の選択であり，オプションとして短縮ボタンが組み込まれることを示している．このような選択がソフトウェアの中に組み込まれることにより，可変性の高いコア資産が形成できる．

このようにユーザ視点での選択が明確になる一方で，フィーチャが多数になるとその組合せは膨大になりフィーチャ設定の工数は増大する．車載ソフトの例では数千ほどのフィーチャにまで及ぶと言われる[24]．そこでフィーチャの論理的組合せを合理化する手法が提案されている[24]．しかし，フィーチャの組合せ数の削減であり，フィーチャ数の削減にまでは踏み込んでいない．また，Schmid[25]はフィーチャ設計にあたってフィーチャを取捨選択する手法を提案している．フィーチャを使うことによるリスクと利益，及びビジネス目標に合致するかを分析し，使用の有無を判定している．しかし，フィーチャ作成コストと利用利益の比較にとどまり，製品固有アプリケーションを作成するコストの検討を含んでいない．

本研究のコア資産分析では分析装置の接続有無を基本フィーチャとして，それを組み込んだ汎用なコア資産とすべきか，組み込まないアプリケーションとすべきかを精

度よく開発前に判定することができる．よって，フィーチャを過度に埋め込むことの複雑性を抑制する．

（３）その他の関連する研究

SPL を用いたときの一般的な経済性は損益分岐点により説明されている[26]．すなわち，SPL におけるコア資産開発費を固定費とし，それを使った製品が一定数以上開発されれば固定費を回収できるとしている．しかし，コア資産を作り過ぎずに開発全体を統制する手法については報告がない．以上のように前述した(課題 1)に対応する製品個別開発を配慮したコア資産統制は行われていない．

既存ソースコードからプロダクトラインを構築する手法では Stoermer ら[27]が提案する MAP がある．アーキテクチャに関する情報を抽出しプロダクトライン化を評価している．しかし，ドメイン知識が評価に含まれず構造の要件に検討余地がある．また，設計書や取扱説明書や専門家などからリバースエンジニアリング情報を得てコア資産を構築する手法も提案されている[28]．情報源にアーキテクチャ要素が含まれているが，それ以上の粒度における解析は述べられていない．アーキテクチャとドメインの両面から再利用性を検討している研究に Koziolek らの研究[29]がある．これはアーキテクトとの面談から抽象レベルの高いアーキテクチャを記述して再構築に活用している．以上いずれの文献もアーキテクチャとドメインの両面をソースコード解析に活用しておらず，従って，前述の(課題 2)のコア資産見積り精度面について言及がない．

コア資産開発と製品開発の全体開発において[30]が WBS の生成を将来の課題としてあげており，前述の(課題 3)に対応する具体的解は書かれていない．

また，マトリックスを用いた SPL 開発としては文献[51]が XDDP(Extreme Derivative Development Process)を SPL 手法に漸次移行するプロセスを報告している．XDDP は製品の追加・変更部位に集中して要求および仕様を分析しコード変更に至るプロセスである．文献では XDDP を SPL に移行するにあたり，要求仕様と実装仕様とのトレーサビリティマトリックスを段階的に明らかにして，コア資産のトレーサビリティに到達することを目標としている．ADM は見積り工程における全体俯瞰

のツールであり，部分開発に活用されるトレーサビリティマトリックスとは異なる．

以上，アーキテクチャ要素とドメイン要素の両面の知識を使っている点では類似例があるが，本研究はこの知識を複数製品ではなく単一製品のソースコード解析に活用してコア資産を抽出している点に特徴がある．単一の製品であっても両知識を用いることにより適切なコア資産の開発計画を導出できる．

２.３　本研究で解決すべき課題

本章では，大きく 2 つに分けて，(1) 処理性能を向上させる自動搬送制御の設計，(2) フレキシブルに分析装置を接続できるプラットフォームの開発手法について，その課題を詳述した．以下，本論文の構成を述べながら，解決すべき課題をまとめる．

2.1 節では，搬送制御の設計にあたり，この問題がオンラインスケジュールとして定式化され，処理性能を重視する設計が必要となることを述べた．

そこで，第 3 章では分析装置に連続して検体が到着しないことによる空きを抑える具体的な搬送制御方式を導出する．この搬送制御方式により分析装置の種類に依存しない形で空きを抑えることが可能となる．また，第 4 章では既に製品化している分析装置 4 台構成までにおける性能を解析している．搬送路における検体搬送能力には，システム処理性能を大きく劣化させないために必要な搬送能力があることを示し，現製品には前提条件があることを示す．そして，この前提条件の下，現製品仕様における処理性能を決定付けるメカニズムを搬送システムのモデルにより明らかにする．このメカニズムにより検体の装置経路ばらつきに対処して空きを減少することができる．第 5 章では，第 3 章で導出した制御方式を用いて，第 4 章で導出した現製品仕様における解析結果を数値実験により検証する．第 5 章では第 4 章の検証に限らず，更に分析装置台数及びバッファ数を現製品仕様より拡大した構成において性能を確認する．

2.2 節では，2.1 節で述べた搬送制御を含めた統合システム全体の開発は，長期に渡るため困難が予想され, 見積り時点でコア資産の統制を行う必要があることを述べた．

これに対して，第 6 章では新たな開発手法である ADM(Architecture Domain Matrix)手法を導いた. コア資産の統制及び見積り精度向上については ADM によるソースコード分析の分解能向上により対処する．また，開発が長期に渡る困難性につい

ては ADM から派生生成される WBS(Work Breakdown Structure)をインタフェースのレビューに活用することで対処する．第 7 章では，第 6 章で述べた手法を実プロジェクトに適用し，その結果を検討し，手法が適切に効果を発揮していることを検証する．また，この第 7 章では，組織面における効用を考察している．

以上の章間の関係を図　２‐１０にまとめる．

	搬送制御の設計	
	位置付け	内容
第3章	制御方式の導出	空きを最小化する方式
第4章	製品仕様 性能解析	前提条件：限界搬送能力
		ばらつき対応のモデル
第5章	第3章の制御による 第4章の性能検証と 拡張仕様	限界搬送能力とモデル
		バッファ数と圧縮率の拡張

	搬送プラットフォーム開発手法	
	位置付け	内容
第6章	開発手法の導出	ADM(Architecture Domain Matrix)と見積もり手順
第7章	第6章の適用と 効果検証	生産性，見積り誤差，コア資産の汎用性

	搬送制御とプラットフォーム開発手法の適用	
	位置付け	内容
第8章	第3章，第4章，第6章を適用した製品	搬送システムの多様な組合せと世界市場状況

図　２‐１０　論文の構成

第３章　搬送制御方式設計

本章では，第 2 章で挙げた搬送制御の課題を受け，自動搬送プラットフォーム実現方式であるサイドトラック方式をパイプライン方式に対する改善方式として提案してその制御方式を導出する．

3.1 節では，搬送プラットフォームに接続される分析装置の機構を概略述べ，検体到着に隙間が発生するとどのように分析装置の処理性能が劣化するかを検討する．3.2 節では，劣化を低減するためのバッファ及び追い越し機能を付加したサイドトラック方式を説明する．3.3 節では，サイドトラック方式の制御方式を導出するとともに，搬送による処理性能劣化が抑えられること，及び，制御方式が分析装置に依存しない仕組みであり容易に分析装置をモジュールとして接続できることを示す．

3.4 節で本章を結論づける．

３．１　分析装置の性能課題

搬送路に接続される分析装置は一定の周期で検体を分注している．その基本的な機構は文献[31]に紹介されており，以下その仕組みを説明する(図　３-１)．

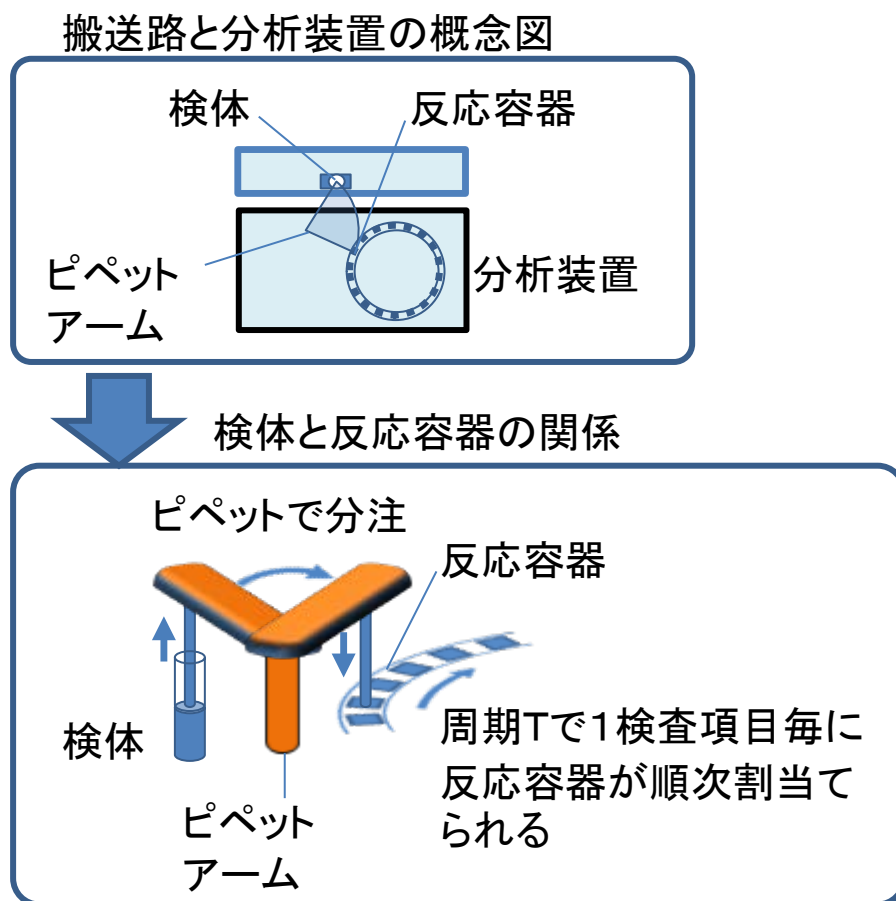



図　3-1　　分析装置における分注機構

分析装置に検体が運ばれると，検体容器内の検体液がピペットにより一部吸い出され，それが分析装置内の反応容器に分注される．分注とは第 1 章で述べたように，検体容器からピペットにより検体を一定量吸い取り，分析装置内の反応容器に吐出する一連動作である．この反応容器には検査項目に応じた試薬が添加され，一定時間，例えば 10 分間，反応させ，反応によって生じる光学上の強度変化を電気変換して濃度値として計算する．分注さえ完了すれば反応容器は反応工程に入るため，その反応の完了を待たずに次の反応容器の分注が行われる．この分注周期 T は例えば 6 秒であればこの場合装置の処理性能は 3600/6=600test/h となる．分析装置内には複数の反応容器が円周状に並び，順次一定の周期で分注位置に移送されては分注が行われる．検体がこの周期に間に合って順次運ばれる限りは，次々と移送される反応容器全てに検体

が分注され分析装置の稼働率は 100%となる．しかし，検体の到着がこの周期に間に合わない場合には，空きの反応容器が生じる．このような遊休の反応容器が多くなると装置が持つ処理性能ポテンシャルを損なうことになる．周期を T とすれば，分析装置の単位時間当りの処理検体数は 1/T(100%処理性能)となるが，N 検体流したときに反応容器の空きが d 回発生すると実質は N/(N+d)だけ処理検体数が劣化(実効処理性能)する(表 ３-１).

表 ３-１　反応容器の空きによる処理性能の劣化

100% 処理性能	実効 処理性能
$\frac{1}{T}$	$\frac{N}{(N+d)\times T}$

T:　サンプリング周期
N:　サンプリングした検査項目数
d:　遊休が生じた周期の数

このような処理性能の劣化を最小限にするために，できるだけ早く連続に検体を搬送して分注を行い，分注を終わった検体もできるだけ早く次の分析装置に搬送する必要がある．搬送システムはこのような要件を満たす必要がある．

３.２　搬送制御プラットフォーム

前述したパイプライン方式は分析装置を搬送路で結合する一般的な方式といえる．しかし，この方式では搬送路沿いに並ぶ分析装置の内，全ての分析装置が分注を終了しないと搬送路を動かせないため，一番遅い分注時間に合わせて搬送路が動く．従って，一番遅い分注を待つ他の分析装置には反応容器の空きが生じ，処理性能が劣化する．

そこで分析装置間で大きな分注時間の差を生まないよう，搬送システムを導入する前には分注時間が均一になるようにシステム設計が行われる．施設で一日に依頼される検査項目の統計値を用いて，分析装置で扱う検査項目種類の数を増減させて分析装

置の分注時間が均一になるように設計する．

このようなシステム設計は負荷分散のために有効で重要であるが，一方で局所的にはやはり分注時間のばらつきが発生する．医療検査を必要とされる人が個別の理由で医療施設にて検査を受けることを考えれば，依頼される検査項目の種類と数が検体毎に独立であると想定できる．独立である以上依頼検査項目のばらつきは不可避であり，このばらつきによる処理性能の劣化を改善する余地がある．

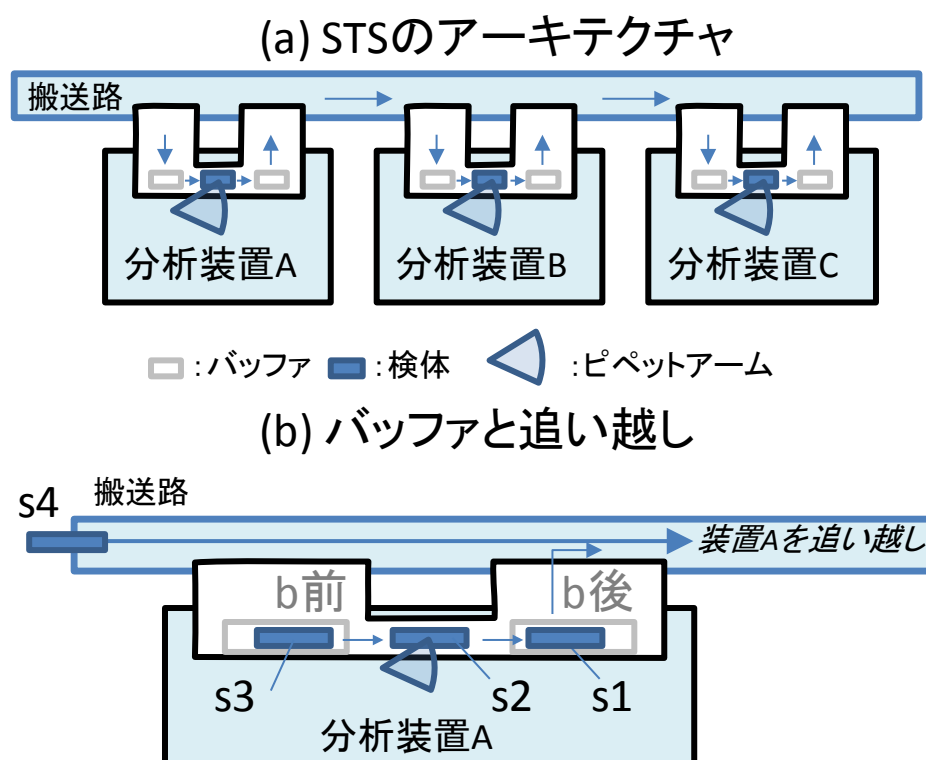



図　３-２　　サイドトラック方式

そこで，検査項目のばらつきに対処できるプラットフォームとして，引き込みラインを有する複数分析装置を１つの共通搬送路に接続したシステムであるサイドトラック方式(STS)が考案された(図　３-２(a))[18].

分析装置内には搬送路から検体を搬入・搬出する経路が用意されている．STS において検体は搬送路入口から任意の分析装置に搬送される．また，任意の２分析装置間

で上流から下流に向けて(図　3‐2(a)の左から右へ)検体を搬送することができる．従って，検体の流れにループが無くデッドロックは起こらない．検査を全部終えた検体は任意の分析装置から搬送路出口に直接搬送される．

PLS が検体を搬送路上に複数縦列させる方式であったのに対して, STS では縦列がない．STS では任意の分析装置間で搬送が必要となった時点で検体が搬送路を占有して搬送が実行される．これは丁度ネットワークバスにおけるデータ伝送と類似している，つまり，検体の搬送が無い限り STS では搬送路上に検体は存在しない．搬送能力が分析装置の処理時間に比べて十分速ければ，共通搬送路を複数の分析装置で競合することは極めて少なく，あったとしても順次実行することによりシステム性能に影響がないことが期待される．

このように任意の分析装置間で搬送が可能となるため，検体の追い越しが可能となる（図(b)の検体 s4)．PLS では搬送路上に検体が縦列している場合，検体は分注が終了しても隣接する分析装置以外には直行できない．STS では搬送路の投入口から必要な分析装置の搬入口に検体が直行する．また，分析装置の搬出口からは任意の下流の分析装置の搬入口に検体が搬送される．縦列する検体が搬送路上で進路を塞ぐことがないため，1 つの検体の遅れが他の分析装置の空きを招くような事態を抑制できる．

STS におけるもう 1 つの特徴はバッファである(図(b))．分析装置には，搬入口と搬出口にそれぞれ最低 1 つのバッファが用意される．検体がこのバッファに搬送されるため，搬送路を常にクリアな状況に維持することができる．加えて，このバッファにより検査項目のばらつきが平滑化される．例えば，図(b)の検体 s1 は分注を終えて b 後の位置にいるが，s2 が分注を終えるまでに s1 が次の装置に出払えば，s2 は分注を終えて b 後に，s3 は分注位置に分析装置の周期に空きを与えず連続して移動する．このように分注位置に遅れて入って分析装置に空きを与えたり，前が詰まって分注位置から出られずに分析装置に空きを与えたりすることをバッファが防止する．STS にはバッファが分注位置の前後に 1 つずつ配置される(図の b 前と b 後)が，本論文では後にこれを増やした場合のシステム処理性能を検討する．バッファ数を増やすことは装置寸法の制限から現実的には難しいので，本論文では数値実験にてバッファ数とシステム処理性能の関係について論じている．

以上のように STS では追い越し機能とバッファにより検査項目のばらつきによる処理性能劣化を抑制する.

３.３　モジュール化のための制御方式

STS の搬送を制御する方式にもばらつきによる処理性能劣化を抑制する仕組みがある．STS では搬送路に検体を搬入する位置に検体が存在するか，搬送路から検体が搬出される位置が空かを常に把握する．この検体有無情報を参照して制御する方式を F/E フラグ方式と呼ぶ．これを模式的に表したのが図 ３-３である.

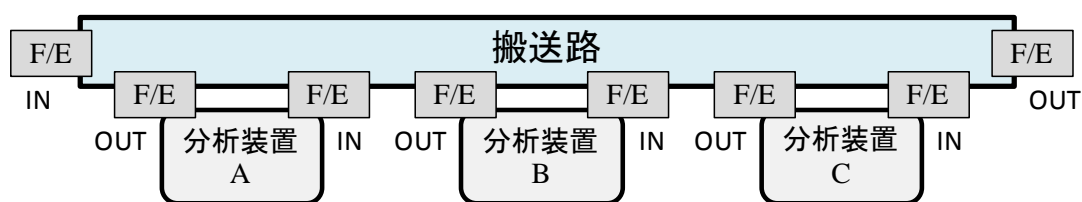



図 ３-３　F/E フラグ方式のシステム模式図

図中 IN と示される位置(IN 位置)が搬送路に検体が搬入される位置, OUT と示される位置(OUT 位置)が搬送路から検体が搬出される位置である. IN 位置と OUT 位置に F/E フラグが配置される．F/E フラグは 1 ビット情報であり，1 ならば Full，つまり検体が存在することを意味し，0 ならば Empty，つまり検体が存在しないことを意味する.

システムでは全 F/E フラグを監視し，以下の 3 条件を満足する i, j の組みを常時走査する.

①IN 位置 i が Full

②IN 位置 i から出る検体の行き先が j

③OUT 位置 j が Empty

このような i,j の組みが見つかりしだい，次の制御を間断なく順次行う.

Step 1: IN 位置 i から共通搬送路に検体を搬入.

Step 2: 共通搬送路を動かして IN 位置 i にある検体を OUT 位置 j の前まで検体を搬送.

Step 3: 共通搬送路から OUT 位置 j に検体を搬出.

この間，共通搬送路は占有される．もしこのような i,j の組合せが複数競合する場合には搬送要求を待ち行列に入れて順次搬送を実行する．

F/E フラグ方式による搬送制御は検体搬送の需要と供給をマッチングさせる形でシステム性能を引き出している．マッチングにおいて時間差があれば待ちが生じる．その状況を図 ３-４に示す.

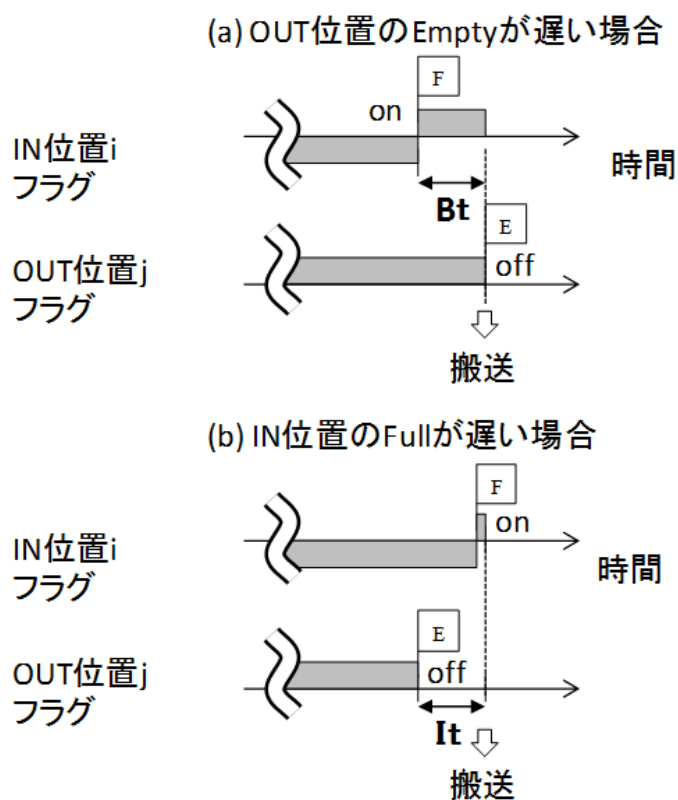



図 ３-４　搬送制御で発生する待ち時間

Full フラグと Empty フラグが立つ時刻の前後関係により２種類の待ちが発生する.

Full よりも Empty が遅い場合(図　3‐4(a))，検体を出す側が行き先の空きを待つ，すなわちブロッキング時間[32](図中 Bt)が発生する．その逆で，Empty よりも Full が遅い場合(図　3‐4(b))，検体を受け入れる側が仕事無く待つ，すなわち空き時間(図中 It)が発生する．

F/E フラグ方式ではブロッキング時間や空き時間が遅延しないよう上述のタイミングをつかんで，搬送を実行する．検体を出す側において検体の行き先は確定しているが，検体を受け入れる側は一般に検体が競合する可能性がある．この競合においては常に一番古く Full フラグが上がった検体が優先搬送されるため, 搬送されずに検体が放置される状態，すなわち飢餓状態が回避される．

本方式以外の搬送制御方式については文献[50]がペトリネットによる搬送系のモデル化を行っている．モデルを実装化して搬送を実現できることが想像される．STS に適用するならばペトリネットを解釈動作させるソフトが複雑となり，F/E フラグ方式の方が簡便であるといえる．

３.４　まとめ

本章では，第 2 章で挙げた搬送制御の課題を受け，自動搬送プラットフォーム実現方式であるサイドトラック方式の優位性を導く制御方式を導出した．

3.1 節では，検体容器から反応容器に検体を小分けする，いわゆる分注動作が周期的に行われているため，この分注周期に間隙を与えない検体の搬送が性能劣化を抑制することを示した．3.2 節では，サイドトラック方式が有するバッファ及び追い越し機能が分析装置の分注周期の空きを抑制することを示した．3.3 節では，サイドトラック方式を支える制御方式を導出した．この制御方式は F/E フラグと呼ぶ空き情報を監視することにより分析装置における空きを検知すると同時に空きに対して到着を望む検体を探索して搬送の遅れによる分析装置の遊休を防いでいる．また，F/E フラグとよぶ簡易なビット情報で分析装置を管理するため，分析装置に依存しない形で空き管理をすることができ，容易に分析装置をモジュールとして接続できる．

以上によりサイドトラック方式の搬送制御を実現する方式が確立された．次章においてはこのサイドトラック方式の性能を具体的に解析する．

第４章　処理性能解析

本章では，追い越し機能が有効に働くために必要な搬送路の搬送能力，PLS に比べて STS の処理性能が向上するメカニズムについて解析する．

4.1 節では，性能比較に用いる尺度(makespan)，及び，PLS 及び STS 両者に投入される検体の列(検査系列)を定義する．4.2 節では，STS の搬送能力にはシステム処理性能を大きく劣化させないために要求される最低ラインの搬送能力，すなわち限界搬送能力が存在することを述べ，限界搬送能力の計算式を示す．

4.3 節では検査系列が搬送システムに投入されると，本来検査系列がもっていた特定分析装置への空きが圧縮される様を解析する．その圧縮幅を示す圧縮率を導入し，圧縮が生まれるメカニズムをモデルとして示す．4.3.1 節では PLS と STS で makespan に差が出てくる様子を事例ベースで明らかにする．4.3.2 節では検査距離という指標を導入する．そして，PLS に比べて STS の makespan が圧縮される率である圧縮率が検査距離とどのような関係になるかを述べる．4.3.3 節では PLS と STS の処理性能を比較するにあたって前提となる条件を設定する．4.3.4 節では，圧縮が発生するメカニズムを搬送システムのモデル化により説明する．搬送システムは空きを圧縮するフィルタとしてモデル化される．

4.4 節で本章の結論を述べる．

４．１　性能尺度と入力検査系列

本研究においてシステム性能の評価尺度として **makespan** を用いる．すなわち，一定の検体列をシステムに投入してから，全検体が分注処理を終えて搬送路出口に出てくるまでの時間を比較する．実際，一日分の全検体がどれくらいの時間で処理されるかは，搬送システムを導入する顧客の重要な関心事となる．この時間を基礎に一日のタイムスケジュールが組まれ，業務効率向上の程度が決まるためである．

2 つのシステムを比較するには同じ依頼の検査項目をもつ検体列(以下，**検査系列**と呼ぶ)を流して makespan を比較する．比較すべき点は，複数の分析装置に跨る検査項目をもつ検体の検査系列を STS が PLS に比べてどのくらい短い時間で処理するかである．ここで，検体が分注されるために経由する分析装置列を**装置経路**と以下呼ぶ．比較に使う検査系列は，①平均として装置の処理時間が均衡していて，かつ，②装置経路がランダムである必要がある．前者①の要件から，分析装置毎で 1 検体が処理される時間を全て同じ時間とする．つまり，1 検体の処理時間は一律に固定の値とする(以後 **1 単位時間**と呼ぶ)．これは，装置当たりの平均処理時間に相当する．

後者②の要件を満たす例をあげれば，2 つの分析装置 A, B が搬送路に接続されている場合，2 つの装置のいずれかで処理されることを前提に装置経路としては 3 通りがある．すなわち，A 単独({A}と表記)，B 単独({B}と表記)，A 処理の後 B 処理({A,B}と表記)である．このような 3 種類の装置経路をもつ検体が一様確率分布に従って到着し，A，B それぞれで 1 単位時間処理されるとすると，局所的には不均衡が発生するものの，平均的には A,B ともに稼働が均一化される．一方，装置経路としては全ての経路を一様に処理していることになり，装置経路の多様性は維持されている．

そこで，図 4-1 に示すような装置経路の組合せを考える．図(a)は分析装置が 2 台の場合の装置経路 3 種{A}{B}{A,B}が示されている．2 進法における 2 ビット列の組合せの内，'00'を除いた構成になっている．本論文において，検査系列はこの図が示すような装置経路をもつ検体がランダムに並んだ列とする．装置数は 2 台から 4 台までを対象とし，図(a)と同様に 3 台構成，4 台構成の装置経路はそれぞれ図(b)，(c)とする．検体数を n 個，分析装置を m 台とすれば，検査系列は，m ビットによって表現される装置経路を並べた長さ n の配列になる．

(a) 2台構成

	A	B
S1	✓	
S2		✓
S3	✓	✓

(b) 3台構成

	A	B	C
S1	✓		
S2		✓	
S3	✓	✓	
S4			✓
S5	✓		✓
S6		✓	✓
S7	✓	✓	✓

(c) 4台構成

	A	B	C	D
S1	✓			
S2		✓		
S3	✓	✓		
S4			✓	
S5	✓		✓	
S6		✓	✓	
S7	✓	✓	✓	
S8				✓
S9	✓			✓
S10		✓		✓
S11	✓	✓		✓
S12			✓	✓
S13	✓		✓	✓
S14		✓	✓	✓
S15	✓	✓	✓	✓

図　４-１　投入する検査系列

４.２　限界搬送能力解析

STS では検体が分析装置間で搬送されるたびに，搬送路は占有される．搬送路は高々1 検体を搬送するために占有されるので，搬送能力が余りに遅ければ搬送時間がシステム性能を劣化させてしまう．そこで，どの程度の搬送能力が許容されるかが課題となる．ここで搬送能力とは 1 検体を運ぶ時間であり，具体的には，検体が搬送路入口または分析装置搬出口から出てから，分析装置搬入口または搬送路出口に至るまでの時間である．

分析装置が m 台あるとすれば，全装置を装置経路とする検体であれば m+1 回の搬送時間を費やし，1 装置を搬送経路とする検体であれば 2 回の搬送時間を費やす(このような回数を以下 **hop 数**と呼ぶ)．検体が n 個あればこれらを n 倍した時間を全搬送

時間として費やす．しかし，STS においては全搬送時間がシステム全体の makespan に加算されるわけではない．

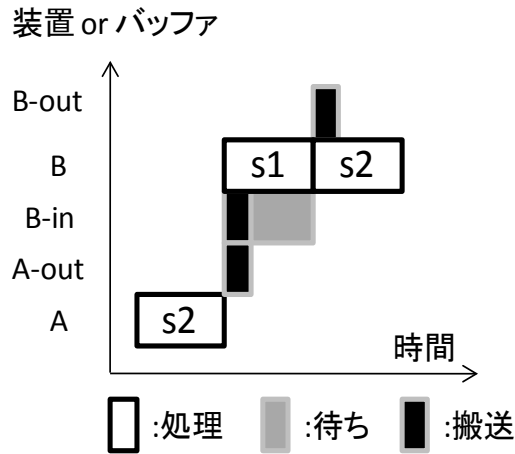



図　4-2　バッファにより隠れる搬送時間

図　4-2 を用いて説明する．図は装置 A に始まり，A の搬出口バッファ(A-out)，B の搬入口バッファ(B-in),装置 B にまで至る経路を検体 s2 が流れる様子を横軸時間のガントチャートで示している．A-out から B-in への搬送は共通搬送路を占有して行われるが，装置 B は検体 s1 の処理中であるため，s1 の処理が終了するまでバッファ B-in 内で検体 s2 は待機する．検体 s1 が終わり検体 s2 は連続的に装置 B 内で処理される．

検体 s2 の経路を時間軸上で眺めると，s2 は s1 の処理時間を待つだけで，A-out から B-in への搬送時間が makespan に加算されていないことが分かる．バッファによって搬送と分注処理が並行に動作することから，この例のように搬送時間が隠れる．STS においては空きを圧縮して検体の密度が高くなるほど，このような並行処理の効果が現れる．

ここで，搬送時間を増やしていくと，どこかの限界を超えた時点で隠すことができなくなり，分注処理時間に空きを生じてしまう．それはバッファによって並行処理される全搬送時間が全分注時間を超えるときである．平均的には以下の式となる．

$$T_{conv} = \frac{T_s}{（検体当りの平均hop数）} \qquad (4.2.1)$$

ここで T_{conv} は限界搬送時間(それを超えるとシステム性能が搬送処理により大きく劣化するような搬送時間．逆数が**限界搬送能力**)，Ts は平均検体処理時間である．これはバッファ数に依存しない．バッファ間およびバッファと分注位置との間で検体が運ばれる時間は，分注周期内に行われることを想定しており，全体時間に影響を与えないためである．

ここで m を搬送路につながる分析装置の数とすると，平均 hop 数は(m+1)を超えない．従って，より厳しい条件(十分条件)としては以下のように表現できる．

$$T_{conv} < \frac{T_s}{m+1} \qquad (4.2.2)$$

実際には図　4‐1 の検査系列から計算すれば，平均 hop 数は m=2 で 2.33，m=3 で 2.71，m=4 で 3.13 となるが，設計として分かりやすい式として，検体処理時間(一単位時間)を(装置数＋1)で除した値を搬送時間の上限としている．例えば，装置が 2 台であれば，平均処理時間の 1/3 が上限となる．装置数を最大 4 台とすると，平均検体処理時間の 1/5 以下で 1 検体を処理すればシステム性能を劣化させない．

この条件はシステム設計の上で好材料となる．すなわち，搬送能力には尤度が与えられている．搬送の対象は液体が入った容器であり，余りに高速にすれば液体がこぼれてしまう．実際に液体容器を高速に搬送する研究もある[33]．尤度があることで，システム性能の劣化を考えないで済む搬送能力設計の目標が設定され，設計自由度が上がる．

4.3　圧縮率解析

4.3.1 PLS と STS の圧縮率差

同じ検査系列を PLS と STS に投入したとき，どのような差が生まれるであろうか．例えば，分析装置 A，B の 2 台構成において，{A}{B}{A}{A}{B}{B}{A}{B}の順で 8 検体の検査系列を投入するケースを考える．列の先頭から検体番号を 1，…，8 に，検体識別子を a1,b2,a3,a4,b5,b6,a7,b8(接頭子 a, b はそれぞれ分析装置 A, B で検査依頼され

た検体を意味する)と名付ける．PLS にこの検査系列が投入されると，PLS の搬送路の A, B の分注位置には，(a1,null), (b2,a1), (a3,b2), (a4,a3), (b5,a4), (b6,b5), (a7,b6), (b8,a7) (null,b8)の検体の組が停止してはシフトする．ここで null は検体が存在しないことを意味する．この内，(b2,a1), (b5,a4), (b8,a7)は A, B いずれの分析装置とも分注を行わない．分注がなくても搬送路の入口出口においてそれぞれ搬入準備，搬出準備が動作するので，その準備時間分搬送路は停止する．分注する検体の組，例えば(a3,b2)では a3 は分析装置 A に，b2 は分析装置 B に分注される．搬送路上の全検体において分注が終了すると搬送路はシフトする．

この動きを図 4‐3(a)にガントチャートで示す．横軸が時間，縦軸には下から搬送路入口(IN)，分析装置 A，分析装置 B，搬送路出口(OUT)の位置で，斜線矩形で搬送路が動作している時間帯，数字入り矩形で分注が行われている時間帯を，示す．矩形内の数字は検体番号である．分注が行われていない検体は空白になっている(例えば，(b2,a1) の b2,a1 や(a4,a3)の a3). この例で makespan は 7.7 単位時間となっている．

一方，同じ検査系列を STS に投入した場合のガントチャートは図(b)になる．

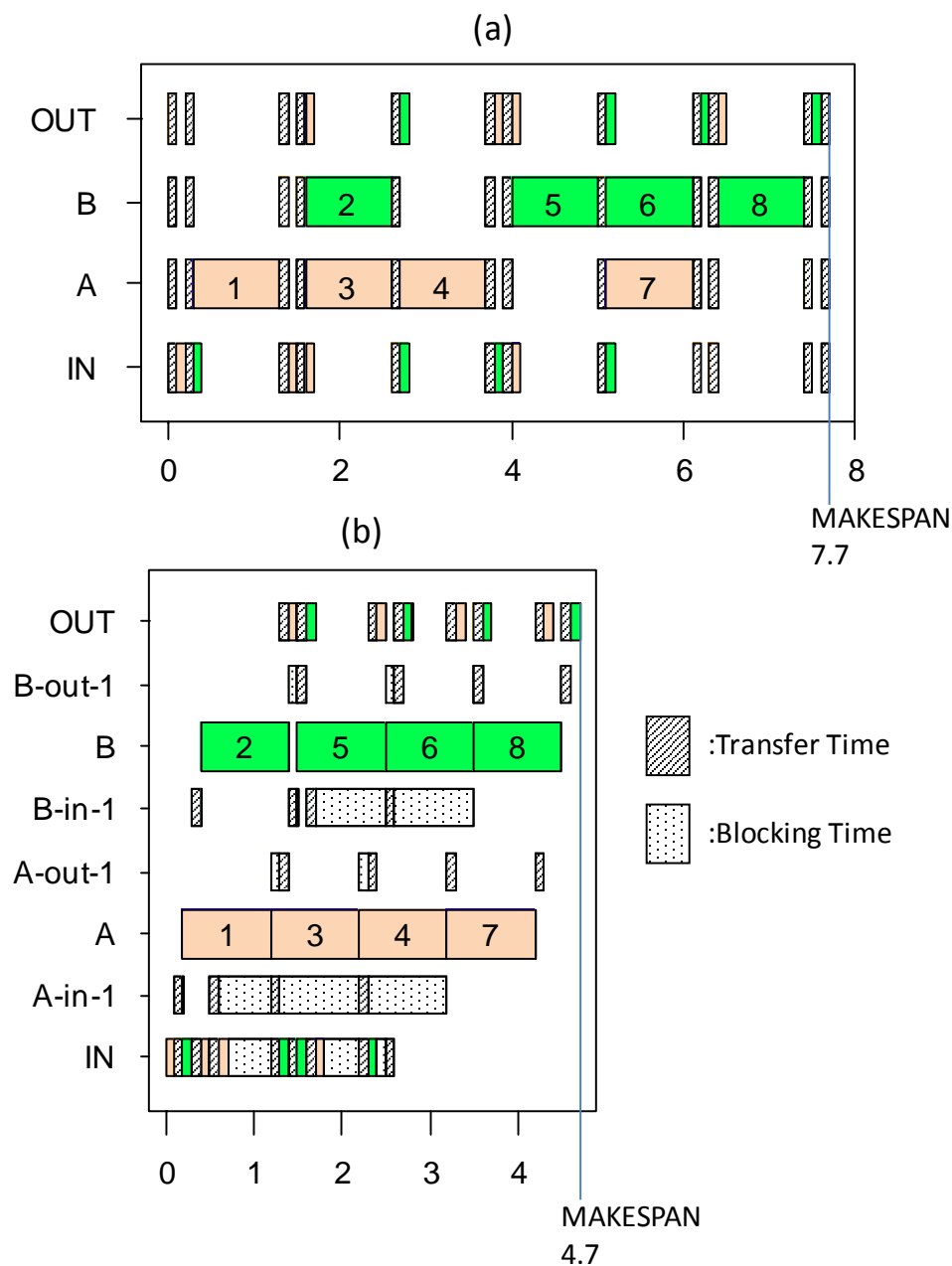



図　４-３　PLS と STS の検体処理動作

この例では分析装置の搬入口・搬出口それぞれに 1 個のバッファ(例えば，分析装置 A の場合, 図中の A-in-1 と A-out-1)が設置されている. 検体 a1 について説明すると, 搬送路入口で搬入準備が行われた後，搬入口(IN)から分析装置　A　の搬入口バッファ

(A-in-1)に検体 a1 は運ばれる．この際，IN，搬送路，A-in-1 は検体 a1 の搬送に占有され， IN と A-in-1 は共に搬送時間を費やす(図中，斜線矩形)．次に，A-in-1 から分析装置 A の分注位置まで検体 a1 は空き時間無しで運ばれる．バッファから装置分注位置への搬送は分析装置の分注周期内で十分に処理できるため，分析装置内の反応容器に空きを生じない．従って，バッファ内の検体は空き時間なく分析装置の処理に移る．但し，装置分注位置に既に別検体が処理中であれば，バッファの検体は当然待たされる(ブロッキング)．検体 a1 は分注処理後，A-out-1 に運ばれるが，ここで一定時間ブロッキングされている(図中，A-out-1 の点描矩形)．これは a4 が A-in-1 に入るために搬送路が丁度占有されているためである．a4 の搬送が終了すると a1 の搬送が行われ，a1 は A-out-1 から搬送路出口(OUT)に運ばれる．a1 は分析装置 B での分注が不要のため，直接出口に送られて追い越しが発生する．このような追い越しにより，出口 OUT に現れる検体の順番は a4 と b5 が投入順番と逆転している．

この例では makespan は PLS が 7.7，STS が 4.7 であり，STS が PLS の 61%であり，従って STS が PLS を 39%圧縮している点で STS は PLS に比べてシステム性能が優れているといえる．実際，チャートで確認すると PLS の(a)では 1 単位時間の空きが A で 2 個，B で 2 個発生しているが，STS の(b)では 1 単位時間の空きは見られない．このような空きの違いが現れる理由は，検査系列における装置経路のばらつきに対して STS のバッファ及び追い越し機能が作用しているためと考えられる．検査系列が{A},{B}の交互繰り返しであれば，PLS でも 2 台の分析装置ともに分注処理を中断する 1 単位時間の空きは発生しない．しかし，装置経路に偏りがある場合，例えば{A}{A}{B}{B}が来た場合には，{A}{A}が来た時点で分析装置 B に空きが生じる．STS は，このような偏りがあっても 2 個の{A}を分析装置 A のバッファ内に取り込み，更に後続の{B}は分析装置 A を追い越して搬送される．一旦バッファに搬送された検体は他の分析装置に干渉されず分注処理される．装置経路のばらつきで発生する空きを STS はバッファ及び追い越し機能により抑制しているといえる．

このように PLS に比べて STS の方が空きを抑制していることが makespan の圧縮につながると考えられる．makespan の圧縮率は 4.1 節で述べた入力系列を用いてシミュレーションにより求めることができる．しかし，より客観的に搬送システムを評価するためには圧縮率が個別の検体系列や検体数に依存しない値であることが望まし

い．そこで２種類のモデル化を行う．１つは搬送システムの稼働結果を空き確率分布でモデル化してそこから得られる空きの特徴量から圧縮率を求める．もう１つは，更に入力系列を空き確率分布でモデル化し，その入力を稼働結果空き確率分布に変換するフィルタに搬送システムをモデル化する．すると入力系列を与えればシミュレーション無しに搬送システム固有の客観的な圧縮率が求められる．以下の節ではこの２種類のモデルを導出する．

４.３.２ 検査距離，圧縮率

システム性能の差は，空き時間の差に起因する．本研究では空き時間情報を引き出すために，検査距離という指標を導入する．そこで，検査距離及び関連する指標について定義する．

検査距離	頻度	確率
1	8	0.615
2	3	0.231
3	1	0.077
4	1	0.077

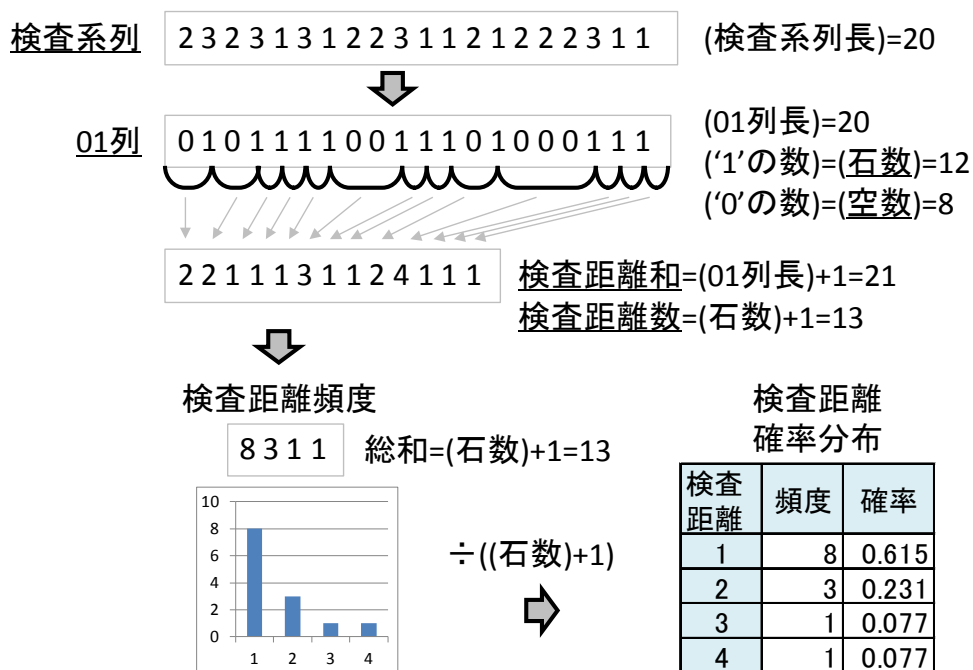



図 4‑4　検査距離と確率分布の定義

図 4‑4 を参照しながら，以下定義する．**検査系列**は検体毎に依頼される装置経路の列である．例えば，分析装置が 2 台の場合には，検査系列の要素である装置経路は 2 ビットの数値で表現され，LSB(Least Significant Bit)が搬送路の最初に接続されて

いる分析装置 A に，MSB(Most Significant Bit)が後続の分析装置 B に対応して，オンであれば経由しオフであれば経由しないことを示す．ここで特定の分析装置に着目して，そこに経由するか否かは，例えば，分析装置 A については LSB が 1 か 0 かで表現され，その列を **01 列**とする．この 01 列中の'1'の数を**石数**，'0'の数を**空数**とする．01 列中，'0'を挟む 2 つの'1'の距離を**検査距離**，その検査距離の列を**検査距離列**と定義する．'1'が隣り合っているならば検査距離を 1 として，'0'を n 個挟んでいれば n+1 とする．01 列の境界は検査距離計算上'1'とみなす．例えば，図の 01 列の最初は'01'で始まるので，境界を'1'とみなすため検査距離は 2 となる．検査距離列における要素の和は 01 列長+1 であり，検査距離列の要素数は石数+1 になっている．検査距離列は**検査距離頻度**にまとめることができ，検査距離数で割ることにより**検査距離確率分布**が作られる．

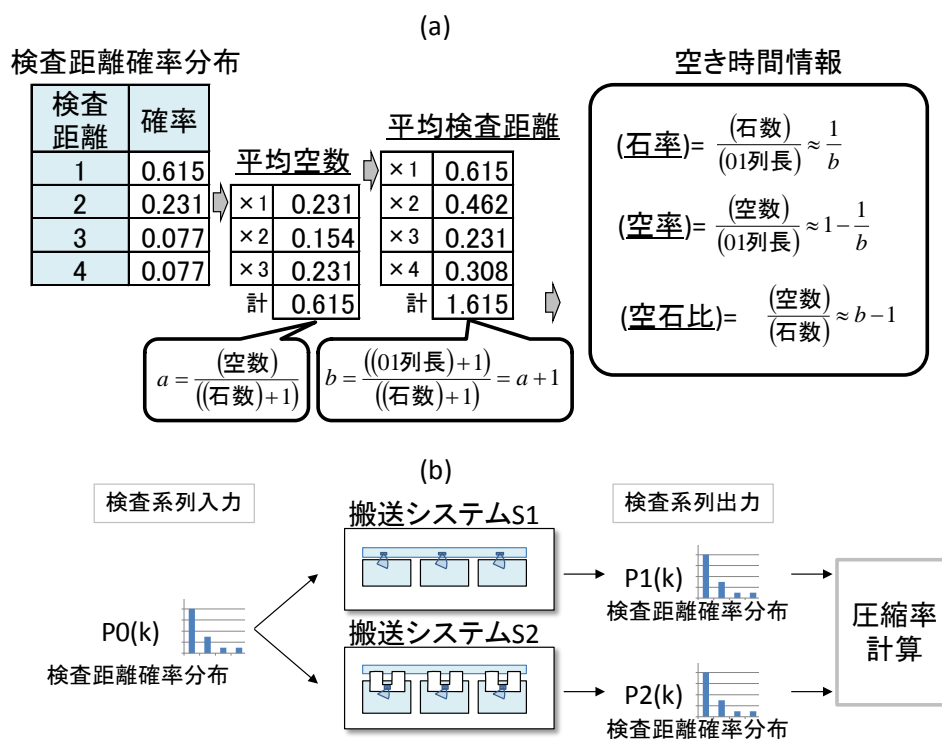



図　4-5　確率分布と空率

以上が検査距離確率分布の導出までの流れとなる．次に，図　4‐5 を使って，確率分布から空き時間情報を導出する．確率分布から**平均空数**(検査距離当たりの空数平均)，**平均検査距離**(検査距離当たりの長さ平均)が求まる．式はそれぞれ図中の式 a, b にあたる．b=a+1 なので，a, b の一方が求まれば他方が計算される．この値により**石率**(01 列内の'1'出現率)，**空率**(01 列内の'0'出現率)が求まる．a, b 式内には+1 が現れて正確には一致しないが，この+1 が無視できる十分に大きい検査系列を扱うため，a, b を利用できる．また，石数に対する空数の割合を**空石比**と定義する．この指標は圧縮率の計算上活用できる．

この空き時間情報から圧縮率を導出する．システム S1 とシステム S2 に同じ検査系列を流し，それぞれの装置で 01 列を観測する．S2 の圧縮率は 01 列長の差が占める S1 の 01 列長の割合となる．

$$
\begin{aligned}
R_{cmp} &= \frac{(\text{S1の}\,makespan)-(\text{S2の}\,makespan)}{(\text{S1の}\,makespan)} \\
&= \frac{(\text{S1の01列長})-(\text{S2の01列長})}{(\text{S1の01列長})} \\
&= \frac{(\text{S1石数}+\text{S1空数})-(\text{S2石数}+\text{S2空数})}{(\text{S1石数}+\text{S1空数})} \\
&= \frac{(\text{S1空石比})-(\text{S2空石比})}{(1+\text{S1空石比})} \qquad (4.3.1)
\end{aligned}
$$

圧縮率 R_{cmp} は，式(4.3.1)により各システムの空石比から導出される．式に至る第 1 式から第 2 式に至る過程で，makespan を 01 列長に入れ替えている．01 列長は装置一台について検査系列を切り出したものであり(図　4‐4)，makespan は装置台数分ある 01 列長の中から一番遅いものの最終時刻になる．従って makespan と 01 列長は厳密には異なるが，検体が各装置を均一に経由していること，よって装置間で 01 列長が均一化していること，により makespan は 01 列長の平均に近くなることを仮定している．第 2 式では S1 石数と S2 石数が現れるが，同じ系列を流しているため相殺される．第 2 式の分母分子を S1 石数で割ることで式(4.3.1)に至る．

以上をまとめると，2 つのシステムにおける検査距離確率分布が分かれば，圧縮率が求まる(図　4‐5 (b))．搬送システムはフィルタのように作用して，検査距離確率分

布を変換する．実際には，分析装置が m 台あれば m 本の 01 列が観測され，その中で一番遅いもの(つまり，一番長い 01 列)が makespan に貢献する．解析的には確率分布を変換するフィルタ式を搬送システムのモデルとして導入する．

4.3.3 前提条件

PLS と STS を比較するために設定する 2 つの前提条件を述べる. 1 つ目の条件として，搬送時間を 0 にする．前述したように，STS では搬送時間をある範囲以内に設定することにより，makespan に搬送時間が大きくは加算されない．一方 PLS では一般に検体数に比例して搬送時間が makespan に加算される．今回の比較では，バッファ及び追い越し機能が寄与する圧縮率をテーマとしている. 過度に STS に有利になることなく，また，比較結果を検体数から独立にするために搬送時間を 0 とする．

2 つ目の条件として，搬送路の入口における搬入準備または出口における搬出準備の時間を 0 とする．PLS では搬送路上の全検体が検査無しの場合，準備時間は makespan に加算され，有限にすれば過度に STS が優位となる．そこで前の条件と同様の理由により，準備時間を 0 とする．ちなみに，この準備時間は，システム全体のスループットに歩調を合わせる必要がある．搬入準備が余りに遅いと，システム内の検体の流れは疎になり空き時間が頻発して本来の性能が出ない．また，搬出準備が余りに遅いと，システム内の検体は出口が塞がれブロッキングされ，やはり本来の性能が出ない．検体の処理時間はどの装置でも 1 単位時間(T)としているので，装置が m 台とすれば準備時間は T/m 以内に設定する必要がある. 準備時間 0 は当然この条件を満たしている．

この 2 つの前提条件により，搬送システムから出力される検査距離が全て整数化(1 単位時間の倍数)され，分布計算が簡易になるという利点も生まれる．

また, 2.1.2 節で述べたように実際のシステムにおいては各分析装置の処理時間が均一になるように，分析装置の種類や検査項目レパートリーをバランス設計している．この実態と同様にシステムに投入される検体が各分析装置で費やす処理時間は平均的にバランスされていることを前提としている．

４.３.４ フィルタモデル

ランダムに検体が投入されるとき検査系列上にはどのような空きが発生し，それはどのような検査距離確率分布を構成するであろうか．解析の端緒として，長さ 3 および 4 の 01 列を考察する(表　４-１).

表の(a)は長さ 3 の，(c)は長さ 4 の 01 列における'1'の検査距離を全ての組合せにおいて一覧している．検査系列の表において，●は境界を意味する．例えば，表の(a)の 4 行目は●110●であり，●1 が距離 1，11 が距離 1，10●が距離 2 と計量されて，検査距離列は 1,1,2 となる．このような(a)(c)の検査距離を頻度集計した表がそれぞれ(b)(d)である．例えば，表(d)は，長さ 4 の検査系列には検査距離 1 が 58%，2 が 25%含まれていることを示す． これは長さが有限であるが，長さが無限または有限でも無限に繰り返されたときの出現確率として以下が導出される.

$$P(k) = (1-q)\cdot q^{k-1} \qquad (4.3.2)$$

P(k)は，01 列中に検査距離 k が出現する確率であり，q は'0'の出現確率である．これは，P(k+1)は P(k)×q であること，及び，P(k)の級数が 1 となることより導かれる．投入する検査系列(図　４-１)を見れば，装置台数によって'0'の出現確率が異なることが分かる．これを反映して，装置毎に検査距離確率関数を表　４-２にまとめる．P のサフィックスは装置台数を示す.

表 4-1 検査距離列の例

(a)

No.	01列 長さ3	検査 距離列
1	●000●	4
2	●100●	1,3
3	●010●	2,2
4	●110●	1,1,2
5	●001●	3,1
6	●101●	1,2,1
7	●011●	2,1,1
8	●111●	1,1,1,1

(b)

検査 距離	個数	%
1	12	60
2	5	25
3	2	10
4	1	5
計	20	

(c)

No.	01列 長さ4	検査 距離列
1	●0000●	5
2	●1000●	1,4
3	●0100●	2,3
4	●1100●	1,1,3
5	●0010●	3,2
6	●1010●	1,2,2
7	●0110●	2,1,2
8	●1110●	1,1,1,2
9	●0001●	4,1
10	●1001●	1,3,1
11	●0101●	2,2,1
12	●1101●	1,1,2,1
13	●0011●	3,1,1
14	●1011●	1,2,1,1
15	●0111●	2,1,1,1
16	●1111●	1,1,1,1,1

(d)

検査 距離	個数	%
1	28	58
2	12	25
3	5	10
4	2	4
5	1	2
計	48	

表 4-2 入力検査系列の検査距離

装置台数	検査距離の確率関数	平均検査距離長	石率	空率	空石比
2	$P_2(k)=\left(\frac{2}{3}\right)\cdot\left(\frac{1}{3}\right)^{k-1}$	$\sum_{k=1}^{\infty} kP_2(k)=\frac{3}{2}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{2}$
3	$P_3(k)=\left(\frac{4}{7}\right)\cdot\left(\frac{3}{7}\right)^{k-1}$	$\sum_{k=1}^{\infty} kP_3(k)=\frac{7}{4}$	$\frac{4}{7}$	$\frac{3}{7}$	$\frac{3}{4}$
4	$P_4(k)=\left(\frac{8}{15}\right)\cdot\left(\frac{7}{15}\right)^{k-1}$	$\sum_{k=1}^{\infty} kP_4(k)=\frac{15}{8}$	$\frac{8}{15}$	$\frac{7}{15}$	$\frac{7}{8}$

以上の確率関数は入力検査系列の性質になる．検査系列がシステムに投入され，オンラインスケジュールされると空きが圧縮されて，確率分布に変化が現れる．その変化の様子を以下説明する．

STS において，バッファは後続の検体のスループットを向上させる．分析装置の搬入口にあるバッファ数を 1 つとすれば，分析装置はバッファと分注位置の 2 箇所に検体を受け入れることができ，両者が空いていれば続けて 2 検体を受け入れることができる．その後 1 単位時間が過ぎればバッファから分注位置に検体がシフトして，バッファは空になり，次の検体を受け入れる．

このことにより，到着する検体間に空きがあっても，分析装置の空きは回避される．図　4‐6 にてその仕組みを説明する．図の(a)(b)ともに 6 検体 a2,b1,b2,b3,b4,a3(接頭子 a, b はそれぞれ分析装置 A, B で検査依頼された検体を意味する)が STS に到着した例である．ここで，図では簡単のため，搬送時間を 0 としている．6 検体の検査系列をみると，分析装置 A から見れば検査系列に 4 検体の空きが生じている．しかし，図(a)のケースでは，分析装置の空きは 1 単位時間に収まっている．これは検体 a2 が到着したときに a1 が稼働中のため a2 はバッファに入ったこと，b1 から b4 が 1 単位時間毎に順次処理されたことが起因している．更に図(b)は，同じ検査系列が投入されて空きが生じない例である．この違いは，分析装置 B が続けて 2 検体 b1,b2 を受け入れたことによる．以上のように，検体系列に分析装置 A に 4 単位時間の空きが生じているにも関わらず，条件によっては分析装置 A の処理の空きは 1 単位時間あるいは 0 となる．STS ではこのように検査系列上の空きを圧縮するという効果がある．

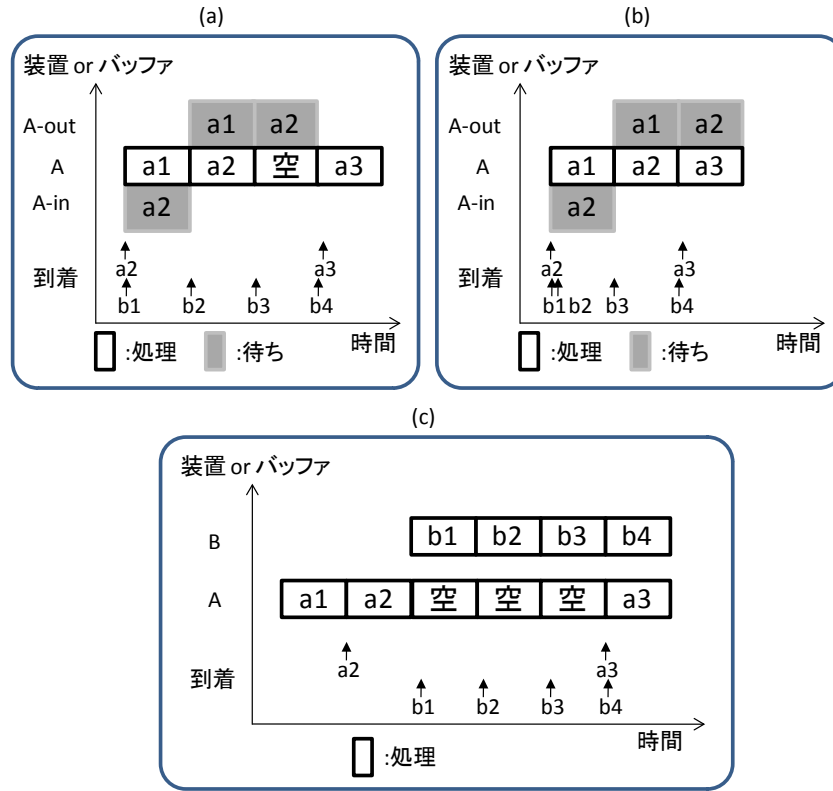



図　4-6　STS と PLS における空き

一方 PLS では，追い越しもバッファも無いため，同じ検査系列を流すと図の(c)が示すように 3 単位時間の空きが生じる．空きが 1 単位時間削減されたのは，最後の工程で b4 と a3 が並列に処理されたためである．

以上から，入力検査系列をシステムに投入すると，検査距離分布における検査距離 2 以上(空きが 1 以上)の頻度が縮退して，少ない検査距離に移動することが推定される．その移動をフィルタ式に表現すると以下 2 式が考えられる．

$$Ps_x(n) = \begin{cases} P_{ORG}(1) + \alpha \cdot P_{ORG}(2) & : n = 1 \\ (1-\alpha) \cdot P_{ORG}(n) + \alpha \cdot P_{ORG}(n+1) & : n \geq 2 \end{cases} \tag{4.3.3}$$

$$Pss_x(n)=\begin{cases}P_{ORG}(1)+P_{ORG}(2)+\alpha\cdot P_{ORG}(3)\cdot & :n=1\\ \left((1-\alpha)\cdot P_{ORG}(n+1)+\alpha\cdot P_{ORG}(n+2)\right) & :n\geq 2\end{cases}\qquad(4.3.4)$$

式(4.3.3)は高々1 個空きが減少するフィルタ式，式(4.3.4)は空きが 1 から 2 個減少するフィルタ式になっている．

(a) 空列slide1

$P_{org}(1)$　$P_{org}(2)$　$P_{org}(3)$　$P_{org}(4)$　$P_{org}(5)$　$P_{org}(6)$ . . .

α　1-α　α　1-α　α　1-α　α　1-α　α　1-α

$Ps_x(1)$　$Ps_x(2)$　$Ps_x(3)$　$Ps_x(4)$　$Ps_x(5)$　$Ps_x(6)$ . . .

(b) 空列slide2

$Pss_x(1)$　$Pss_x(2)$　$Pss_x(3)$　$Pss_x(4)$　$Pss_x(5)$　$Pss_x(6)$. . .

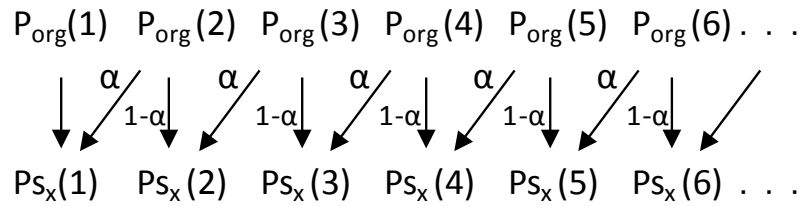
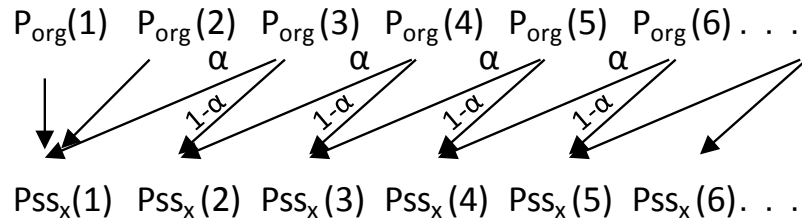



図 4-7 フィルタの構造

これらを模式的に描いたのが図　4-7 である. 前者を空列 slide1, 後者を空列 slide2 と名付ける．式中，$P_{org}(n)$は入力検査系列における検査距離分布を示す．空列 slide1 におけるパラメータ α が 1 のフィルタと空列 slide2 のパラメータ α が 0 のフィルタは等価である．空列 slide1 におけるパラメータ α が 0 であればフィルタは入力列を何も変更しない. 空列 slide2 のパラメータ α が 1 であれば, 空列は 2 段跳びでシフトする．搬送システムの出力検査距離確率分布に対して上式がフィットするようにパラメータを調整することで，搬送システムはフィルタ式として抽象化される．

4.4 まとめ

本章では，搬送路の限界搬送能力及び，PLS に比べて STS の処理性能が向上するメカニズムについて解析した．

4.1 節では，性能比較に用いる尺度を検体列がシステムに投入された時点から全ての検体が処理を終えるまでの時点までの時間，すなわち makespan を尺度とした．検体はその依頼検査項目に従っていくつかの分析装置に立ち寄り検査を行う．この立ち寄りのパターンを装置経路と呼び，装置経路の全ての組合せをランダムに並べたものを検査系列とした．各装置で検体が処理される時間は一定とした．これにより標準的なばらつきある検体を表現している．

4.2 節では，STS において搬送時間が検体処理時間以下であれば，搬送先の検体処理を乱さず並行してバッファに取り込まれるため，システム全体の makespan に加算されないことを明らかにした．逆に加算されてしまう程度に大きい搬送時間は検体処理時間を((装置数)+1)で除した値を超える点であることを示した．これは今後の搬送設計に有益な尤度基準を与えている．

4.3.1 節では PLS では空きが発生している検体列を STS に流すと空きが減少している事例を述べた．この事例によりバッファや追い越し機能がこの空きの減少に貢献していることを示した．4.3.2 節では特定の分析装置に検体が到着する間隔を検査距離と定義し，その関連する空き情報指標を定義した．その内の 1 つである空石比(検査依頼数に対する空きの割合)を用いると 2 つの搬送システム間の makespan の圧縮率が計算できることを示した．4.3.3 節では，PLS と STS の処理性能を比較するにあたって搬送時間，搬送路入口及び出口における準備時間を 0 に設定することを述べた．この設定は PLS と STS を比較にするにあたり過度に STS が優位にならない配慮，及び，比較した結果が検体数に依存しないようにする配慮によることを示した．4.3.4 節では，長い検査距離が搬送システムに入って出力されるとより低い検査距離にシフトしていることに注目している．これを抽象化して搬送システムの入力である検体列における検査距離確率分布が搬送システムの出力における検査距離確率分布にフィルタ変換されるというモデルを導入した．このモデルが空きを圧縮するメカニズムとなる．

以上により，4.3 節では，PLS と STS に検体を流したときの事例を観察し，検査距離確率分布という検査列の特徴量抽出を行っている．その結果，搬送システムは検査距離確率分布のフィルタとしてモデル化できることを述べた．つまり，ある検査距離確率分布の検体列を搬送システムに入力すると，その出力である検査距離確率分布は検査距離の長い部分が縮退して検査距離の短い部分が増える．

また，2 搬送システムから出力される検査距離分布の検査距離期待値から圧縮率が計算されることを示したことも意義深い．従来は，シミュレータで個別検査室の平均的な検体列を流して，個別に処理性能を調べていた．しかし，このモデルを使うことで計算式によって性能を予測できる可能性が出ている点で有益なモデルであるといえる．また，フィルタ式は搬送システムの処理性能特性をモデル化しているため，搬送システムを解析的に比較する可能性がある点でも意義がある．

第５章　処理性能検証

本章では，数値実験に用いたシミュレータの説明を行った上で，前述した解析結果を数値実験により検証・考察する．加えて装置台数及びバッファ数が既存の製品仕様を超えるときの性能について実験し考察する．

5.1 節では，PLS 及び STS のシミュレータを概説する．5.2 節では STS の限界搬送能力が存在すること及び解析結果である限界搬送能力の計算式に実験結果が当てはまることを示す．5.3 節では解析で述べたモデルに基づき検査距離情報から算出した makespan の圧縮率と直接 makespan から算出した圧縮率が符合することを示してモデルの確からしさを示す．同時に装置台数が 2 から 4 までの構成において圧縮率を数値で示す．5.4 節では STS のバッファ数を 1 から 3 までに増やし，装置台数も 4 から 8 に増やすことにより圧縮率がどのように変化するかを実験して論ずる．

5.5.節で，本章を結論づける．

５.１　シミュレータの構成

数値実験に用いるシミュレータの構成を概略述べる．検体数を n 個，分析装置を m 台とすると，入力検査系列は m ビットから成る装置経路を n 個並べた待ち行列 Q になる．PLS のシミュレータでは以下のステップを実行する．

(Step 1) 搬送路に見立てた長さ m の配列を用意して，その配列インデックスを分析装置に対応させる．

(Step 2) 待ち行列 Q から先頭 m 個の要素(装置経路)を取出し配列に割り当てる．

(Step 3) 搬送路配列の全インデックスにおいていずれかの装置経路が i 番目の装置を含んでいれば，時間を 1 単位時間更新する．

(Step 4) 搬送路配列を１要素分シフトし，待ち行列Ｑから１要素取出して，シフトして空いた配列に割り当て(Step 3)に戻る．待ち行列Ｑが空であれば終了する．

STS のシミュレータについては，STS の制御方式(前述)に従って制御を実行する．装置は，遊休，分析中，ブロッキングの 3 状態の遷移によってモデル化される．また，バッファは，遊休，ブロッキングの 2 状態でモデル化される．

以上の 2 シミュレータには，図 4-1 に基づいてランダムに発生された入力検査系列が投入される．出力は makespan(全検査系列の終了時刻)と各装置における空き情報になる．シミュレータの動作結果は PLS/STS のいずれも GANTT チャートに出力され，視覚的に動作を確認することができる．図 5-1 に 20 検体を PLS 及び STS に投入したときにシミュレータから出力された GANTT チャート(上図が PLS、下図が STS)を示す．4.3.3 節の前提条件を満たしているため，makespan に端数がなく検体の単位時間の整数倍になっていることが分かる．

本シミュレータでは,ハード機構がモデル化され個々の機構は省かれている.また,検体の処理時間を 1 単位時間としてその相対時間として搬送路の処理時間や makespan を扱っている．従って，個々の機構に対する要求や絶対時間的な要求については導出することはできない．一方，限界搬送時間は検体の処理時間の比から求めることができるが，最低の検体処理絶対時間を考慮することにより絶対時間で限界搬送時間を表わし設計基準とすることができる．また，PLS との比較により算出される STS の空き圧縮率は比率であり，相対時間により導出可能となる．このようにシミュレータでは相対時間評価を前提とするが,それでも有意義な結果の算出が可能となる．

なお，シミュレータは統計処理用言語[34]を用いて実装している．

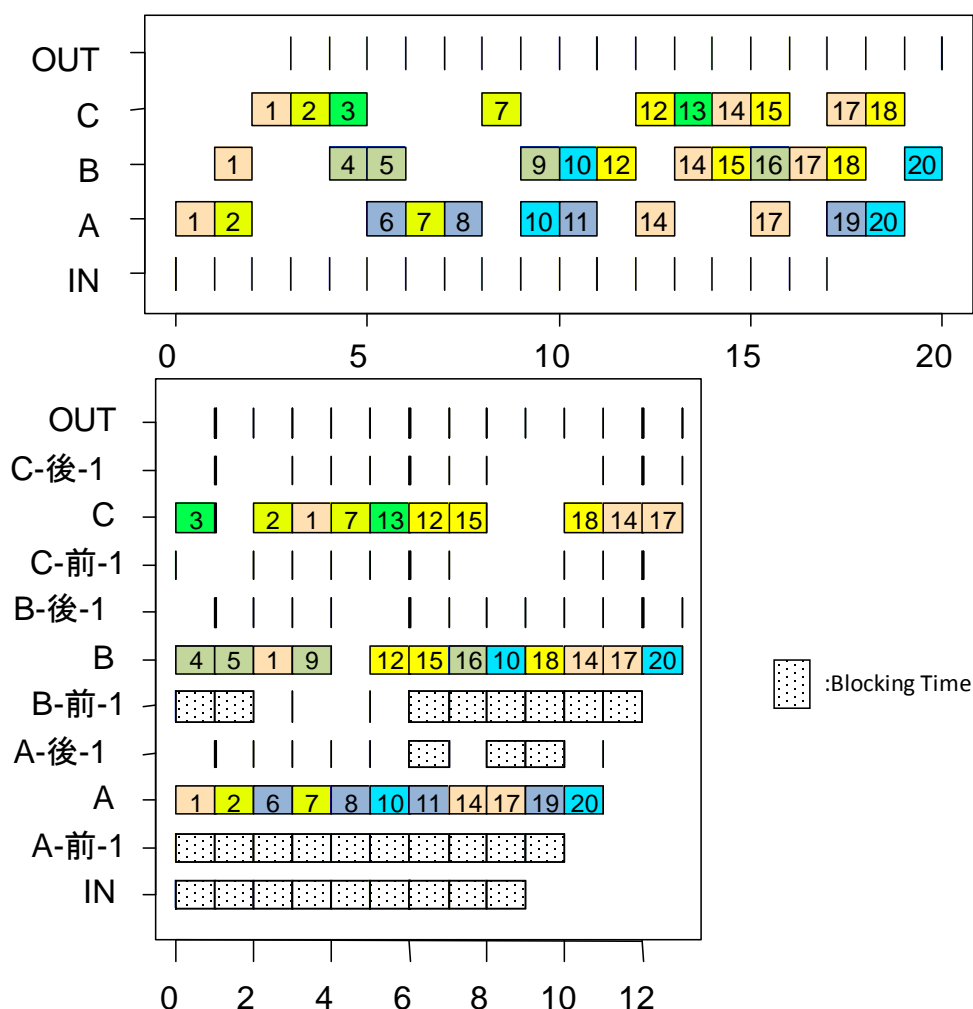



図 5-1 GANTT チャート例

５.２ STS の限界搬送能力

図 5-2 に実験で得た，搬送能力と makespan の関係を示す．装置台数 2, 3, 4(図中 m=2 などの表示)のそれぞれに関係曲線を描いている．投入した検査系列の長さは 200, 搬送能力として検体 1 個当たりの搬送時間を 0 から 1(1 単位時間)まで 0.01 刻みに変化させている．それぞれの搬送時間毎に 20 回シミュレータを動作させ，20 点の makespan の平均を図にプロットしている．

図から分かるように，搬送時間は 3 曲線ともに搬送時間 0.2 から 0.3 当たりを境(CP:

Critical Point)にして，傾きが大きく変化している．つまり，この限界搬送能力より低い(限界搬送時間より長い)と makespan が大きくなりシステム性能が劣化している．解析結果では，この CP の値が 1/((装置数)+1)であると結論したが，曲線では必ずしも CP が明らかではない．そこで，搬送時間の前半 0 から 0.15 まで，及び後半 0.65 から 1 までのそれぞれに回帰直線を求め，その交差点として CP を計算した．その結果を限界搬送能力として表 5‐1 に示す．

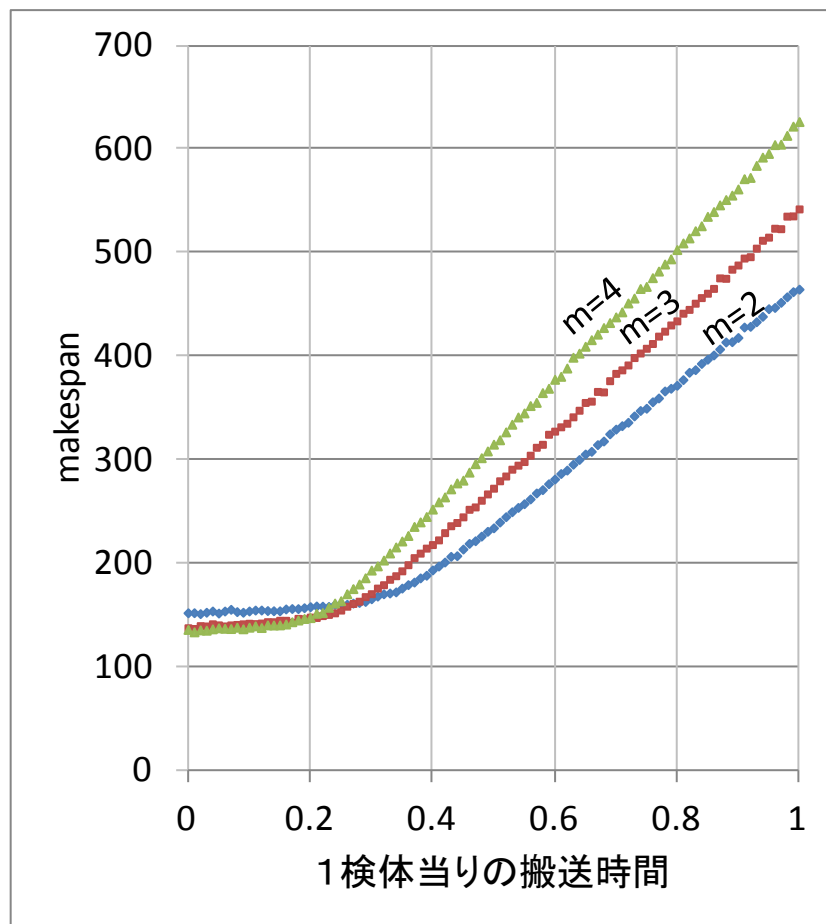



図 5‐2 搬送能力と makespan

表 5-1　限界搬送能力

	回帰直線1(ax+b)		回帰直線2(ax+b)		実験CP	解析CP 1/((装置数)+1)
	a	b	a	b		
m=2	17.0	152	461	5.75	0.331	0.333
m=3	40.7	138	538	4.21	0.269	0.250
m=4	37.0	135	625	1.60	0.227	0.200

表には回帰直線 1(前半)，2(後半)の傾き a と切片 b，これらが交差する点から求めた実験 CP，解析結果による解析 CP が記されている．回帰直線 1 が若干の傾きをもっているが，これは検査系列を実行したときにできる空き時間個数に比例している．検体が装置に到着して空き時間が解消される直前には搬送時間が makespan に反映されるためである．また，回帰直線 2 の傾きは，(平均 hop 数)×(検体数)に比例する．また，バッファ数を 2 にして同実験を行った結果，回帰直線 1 の傾きが減少(空き時間個数の減少による)が確認された他は表 5-1 と優位な差が無く，バッファ数と限界搬送時間とは独立であると考える．

実験 CP と解析 CP は近い値を示している．搬送機構の設計条件として，最大 4 台構成を想定すれば，平均検体処理時間(1 単位時間)の 1/5(0.2)以内であればシステム性能が劣化しないという結果が得られた．この実験結果は，解析結果と一致する．

5.3　PLS に対する STS の圧縮率

図 5-3 に PLS/STS に検査系列を投入した結果を示す．

図は検査距離の 1 から 4 までを抜粋して装置台数構成毎にグラフ化している．ORIGIN と表記のあるグラフは入力系列の分布，PLS/STS と表記のあるグラフは PLS/STS の出力系列の分布である．検査系列の長さは 400，いずれの装置構成も 40 回投入しており，出力された全検査距離頻度分布を加算した後全頻度数で割り確率を求めている．PLS/STS ともに，搬送時間及び搬送路の搬入準備時間及び搬出準備時間は 0 である．STS のバッファ数は 1 である．

	1	2	3	4
ORIGIN	0.670	0.222	0.071	0.025
PLS	0.780	0.147	0.048	0.017
STS	0.905	0.073	0.014	0.005

(b) m=3

	1	2	3	4
ORIGIN	0.572	0.246	0.105	0.045
PLS	0.630	0.225	0.088	0.035
STS	0.864	0.100	0.023	0.008

(c) m=4

	1	2	3	4
ORIGIN	0.533	0.251	0.115	0.053
PLS	0.563	0.245	0.105	0.047
STS	0.837	0.115	0.031	0.010

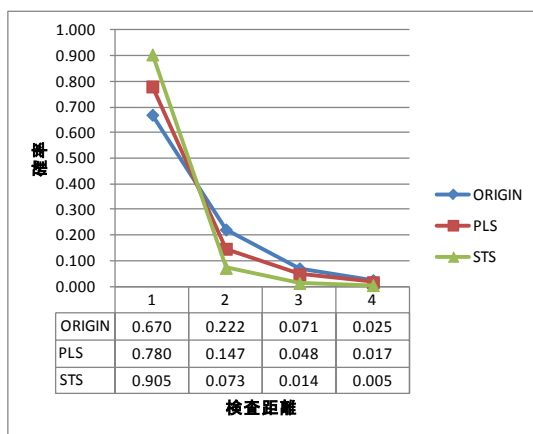
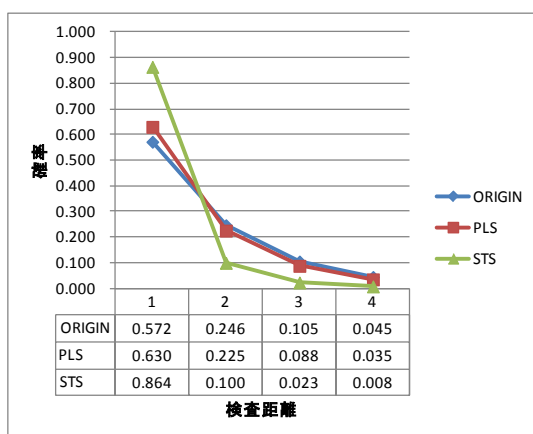
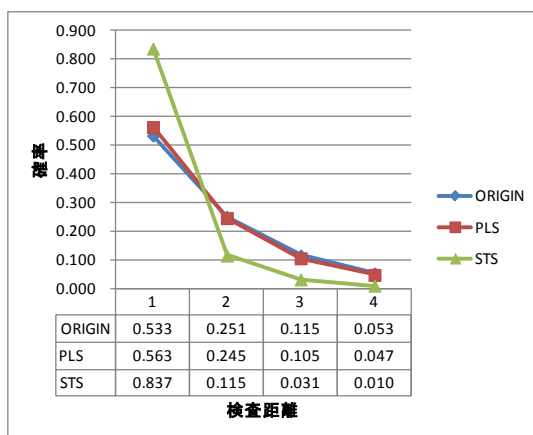



図　５-３　検査距離分布出力

図のグラフによると，いずれの分布も，入力(ORIGIN)に対して，PLS, STS の順に検査距離 1(空き 0)の比率が向上し，検査距離 2 以上(空き 1 以上)の比率が低下している．これは空きの圧縮の結果である．グラフ下段の ORIGIN の確率分布を見ると表４‐２に示した確率分布に近いことが分かる．

ORIGIN の確率分布から PLS/STS の確率分布を得るフィルタ式を検討した結果，解析で導出した空列 slide を改良した形となった(式(5.3.1), (5.3.2))．

$$P_{PLS}(n)=\begin{cases} P_{ORG}(1)+\alpha\cdot P_{ORG}(2) & : n=1 \\ (1-\alpha)\cdot P_{ORG}(2)+\alpha\cdot P_{ORG}(3)+\beta\cdot P_{ORG}(3) & : n=2 \\ (1-\alpha)\cdot P_{ORG}(3)+\alpha\cdot P_{ORG}(4)+\beta\cdot P_{ORG}(4)-\beta\cdot P_{ORG}(3) & : n=3 \\ (1-\alpha)\cdot P_{ORG}(n)+\alpha\cdot P_{ORG}(n+1)+\beta\cdot P_{ORG}(n+1)-\beta\cdot P_{ORG}(n) & : n\geq 4 \end{cases} \tag{5.3.1}$$

$$P_{STS}(n)=\begin{cases} P_{ORG}(1)+P_{ORG}(2)+\alpha\cdot P_{ORG}(3) & : n=1 \\ \left((1-\alpha)\cdot P_{ORG}(3)+\alpha\cdot P_{ORG}(4)\right)\cdot\left(1+\beta\right) & : n=2 \\ \left((1-\alpha)\cdot P_{ORG}(4)+\alpha\cdot P_{ORG}(5)\right)\cdot\left(1+\beta\right)-\beta\cdot(1-\alpha)\cdot P_{ORG}(3) & : n=3 \\ \left((1-\alpha)\cdot P_{ORG}(n+1)+\alpha\cdot P_{ORG}(n+2)\right)\cdot\left(1+\beta\right)-\beta\cdot P_{ORG}(n) & : n\geq 4 \end{cases} \tag{5.3.2}$$

このフィルタ式は空列 slide に対してパラメータ β を付加している． β が 0 の時は丁度空列 slide と一致する． β により，空列のシフトを加速している．

表 ５‐２に圧縮率の結果を一覧する．表(a)は式(5.3.1)(5.3.2)に ORIGIN の確率分布を与えて，PLS/STS の確率分布にベストフィット(誤差の二乗和が最小化)するよう α と β を調整した結果である．それぞれの空石比から式(4.3.1)により圧縮率を求めている．

表(b)はシミュレータに検査系列を投入して得られた確率分布から空石比及び圧縮率を計算している．また，表(c)は別途検査系列長 400 を 40 回シミュレータに流して，空石比を使わず，直接 makespan から圧縮率を求めている．

表(a)(b)(c)の圧縮率はかなり近い値を示しており，その差は表(c)の標準偏差(SD)内に収まっている．この点で，フィルタ式(5.3.1)(5.3.2)は PLS/STS をモデル化していること，つまり解析にて述べたように，搬送システムがフィルタ式としてモデル化されていることを示す．また STS は PLS に比べて 4 台構成において処理時間を 30%圧縮(表(a)(b)内 0.306，表(c)内 mean(平均)0.308)することが分かった．

表 5-2　圧縮率

(a)

	PLS			STS			圧縮率
	alpha	beta	空石比	alpha	beta	空石比	
m=2	0.50	0.01	0.328	0.18	0.16	0.128	0.151
m=3	0.23	0.10	0.627	0.43	0.26	0.193	0.267
m=4	0.11	0.08	0.804	0.42	0.30	0.253	0.306

(b)

	PLS 空石比	STS 空石比	圧縮率
m=2	0.330	0.132	0.149
m=3	0.608	0.195	0.257
m=4	0.784	0.238	0.306

(c)

	PLS makespan		STS makespan		圧縮率	
	mean	SD	mean	SD	mean	SD
m=2	355.8	5.79	302.8	6.64	0.149	0.014
m=3	369.0	6.13	275.4	7.98	0.253	0.018
m=4	383.5	3.93	265.4	5.29	0.308	0.015

5.4　STS のバッファ数とシステム性能

前節と同じ条件でバッファ数を 2 と 3 に変更して実験した結果を表 5-3 に示す．

バッファ数が bn とは，分析装置の搬入口，搬出口それぞれにバッファが bn 個存在することを意味する．表中，bn=2, bn=3 の列がその結果であり，PLS 及び STS の bn=1 の列は前節表　5‑2 の再掲値である．

バッファ数が増えると一般に圧縮率が高まる．入口のバッファが 1 つ増えることにより 1 つ余分に検体を取り込めるので，ゼロ時間で分注位置に運べる検体が増え，空き無く分析することができる．

表　5‑3　バッファ数と圧縮率

		PLS	STS		
			bn=1	bn=2	bn=3
m=2	空石比	0.330	0.132	0.085	0.066
	圧縮率	−	0.149	0.181	0.196
m=3	空石比	0.608	0.195	0.128	0.100
	圧縮率	−	0.257	0.296	0.315
m=4	空石比	0.784	0.238	0.157	0.121
	圧縮率	−	0.306	0.356	0.364

空石比を見ると，バッファ数が 2 以上になると 0 に近い値を示している．空石比は，検体数(石)に対する空きの割合に相当するから，空石比が 0 に近いとは空きが 0 に近いことを意味する．従って，圧縮率もバッファ 3 個ではバッファ 2 個に比べて大きな効果が得られていない．また，バッファ数 2 もバッファ数 1 の効果ほどの圧縮率向上は見られない．

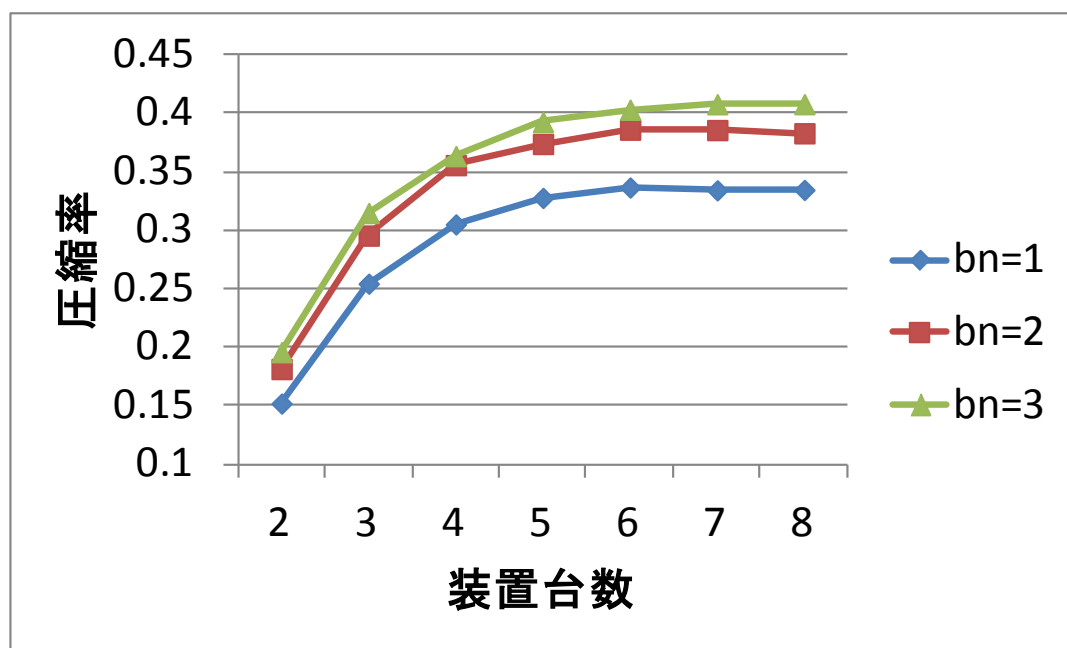



図　5‐4　装置台数とバッファ数と圧縮率

図　5‐4 に装置台数を 8 台まで増やして，バッファ数が 1 から 3(図中，bn=1 から bn=3)まで圧縮率をプロットしたグラフを示す．全ての台数において bn=2 と bn=3 では圧縮率の大きな改善はない．装置 8 台構成では bn=1 で 33%，bn=2 で 38%で，5 ポイントの差は装置 4 台構成と変わらない．

まとめると，バッファ数 3 つは 2 つに比べて効果を期待すべきではなく，バッファ数 2 つは 1 つに比べて圧縮率を 4 台構成以上で 5 ポイントほど改善する．

一般に，分析装置の巾寸法による限界からバッファの数を多くすることは設計上難しい．バッファ増設によるインパクトが推定できることで，バッファ数設計を助ける．バッファ 1 個で十分な圧縮率を達成できるとも評価できる．

5.5　　まとめ

前章で求めた搬送システムの性能解析結果を実験により検証した．また，既存製品の仕様を超えたときの性能についても論じた．

5.1 節では，数値実験に用いる PLS 及び STS のシミュレータを概説した．特に STS では第 3 章で論じた制御方式を用いてシミュレータを構築している．また，前章で設定した前提条件のため，makespan が検体処理時間の整数倍になることをシミュレータ出力の GANTT チャートで示した．5.2 節では限界搬送能力の存在とその値を数値実験により求めた．既存製品仕様である装置台数 2，3，4 について搬送能力(1 検体当たり搬送時間)を変えながら makespan を測定したところ，確かに限界搬送能力が存在することが明らかになった．限界搬送能力以内であればほとんど makespan が一定であり，それを超えるとリニアに makespan が増加する．この限界搬送能力の測定結果は解析結果である検体処理時間の 1/((装置数)+1)とほぼ一致した．製品仕様の装置台数は最大 4 台であるため，検体処理時間の 1/5 以下であることがシステム性能を劣化させない条件であることが示された．これによって，現在製品仕様における搬送能力の限界が明らかになった．

5.3 節では装置台数 2，3，4 について PLS および STS の出力結果における検査距離頻度分布を数値実験で得た．この分布結果から，入力となる検体の検査距離頻度分布が第 4 章で求めた指数分布に従うこと，PLS および STS ともに出力の検査距離頻度分布で検査距離の高い部位が低い部位にシフトする状況が確認できた．第 4 章で得たフィルタ式に改良を加えた式において検査距離頻度分布を近似したところ，この近似検査距離頻度分布から算出した PLS に対する STS の圧縮率(a)が元々の検査距離頻度分布から算出したもの(b)，及び，直接 makespan から算出したもの(c)と符合した．これにより，フィルタ式が搬送システムをモデル化できていること，フィルタの入出力である検査距離頻度分布が処理性能を算出する基礎となっていることが示された．また，製品仕様である装置台数 4 において，STS は PLS に比べて処理時間を 30%圧縮することが分かった．この評価値は検体数に依存しない点で純粋に検体の装置経路ばらつきに対する圧縮率を示している．従来であれば，検査室毎に過去の検体列を投入してシミュレーションで性能をチェックしていた．このような具体的な数値で客観的に性能比較ができたことは今後の搬送システムの設計において大きな道標となる．

5.4 節では，製品仕様であるバッファ数 1 及び装置台数最大 4 を超える仕様において処理性能を数値実験で求め検討した．バッファ数を 3，装置台数を 8 まで増やして PLS に対する STS の圧縮率を求めた．その結果バッファ数が 2 と 3 では大きい圧縮

率の向上は見られず，バッファ数１と２の間では装置４台構成以上で５ポイントほど改善する程度であり，分析装置内の搬入口・搬出口にあるバッファ数は２個を超えると大きい圧縮率の改善が得られないことが分かった．つまり，バッファ数１で十分な圧縮率が得られることになり，この結果は今後の搬送システムにおけるバッファ設計に役立つ．一般にバッファの数を増やすことは，装置スペースまた装置コストの上で難しい設計となる．バッファ数１つで十分な圧縮率が得られることが分かり，STSの設計上の優位点が見出された．

また，本章のSTSとPLSの比較実験では，第４章の前提条件に基づいて搬送時間を０に設定している．もし現実に合わせて搬送時間を有限にすれば，PLSは検体数に比例して搬送時間が処理時間に加算されるのに対して，STSは搬送時間が限界搬送時間内であれば搬送時間は処理時間に加算されない. 従って, 有限な搬送時間の元では, STSは更に優位となる．

第６章　開発手法: ADM 手法

本章では，前章までに述べた自動搬送プラットフォームをコア資産としてどのように開発するべきかを述べ，開発手法として ADM 手法を提案する．

6.1 節は SPL におけるコア資産の見積り手法として ADM 手法を提案する．6.2 節ではコア資産とアプリケーションとのバランスを統制することにより理想のコア資産が生まれることを示した上で，AIMS において分析装置を接続した場合にアーキテクチャに影響を与える範囲を示す．6.3 節では ADM 手法の特徴であるアーキテクチャ要素とドメイン要素の 2 軸からなる表を導入して，アプリケーションとバランスを取りながらコア資産を抽出する手法を述べ，6.4 節でその手順をステップ毎に説明する．

6.5 節では本章を結論付ける．

６.１　SPL における位置付け

SPL のエンジニアリングは，ドメインエンジニアリング(または Core Asset Development)，アプリケーションエンジニアリング(または Product　Development)，マネジメントの 3 活動で定義されている[35]．

この SPL の 3 活動に基づいた開発の流れを図　6‑1 に記す．ドメインエンジニアリングでは，将来いつごろにどのような新製品を顧客に提供すべきかといった事業戦略が入力となり，分析装置のドメインにおいて維持すべきコア資産を出力する．アプリケーションエンジニアリングでは，コア資産を有効に適用しながら事業戦略に基づき顧客満足を最大限に提供できる製品を開発する．両エンジニアリングに渡り一貫した

経営資源の配置などのマネジメントが欠かせない．本論文は，ドメインエンジニアリング及びアプリケーションエンジニアリングに跨る開発手法を提案している．

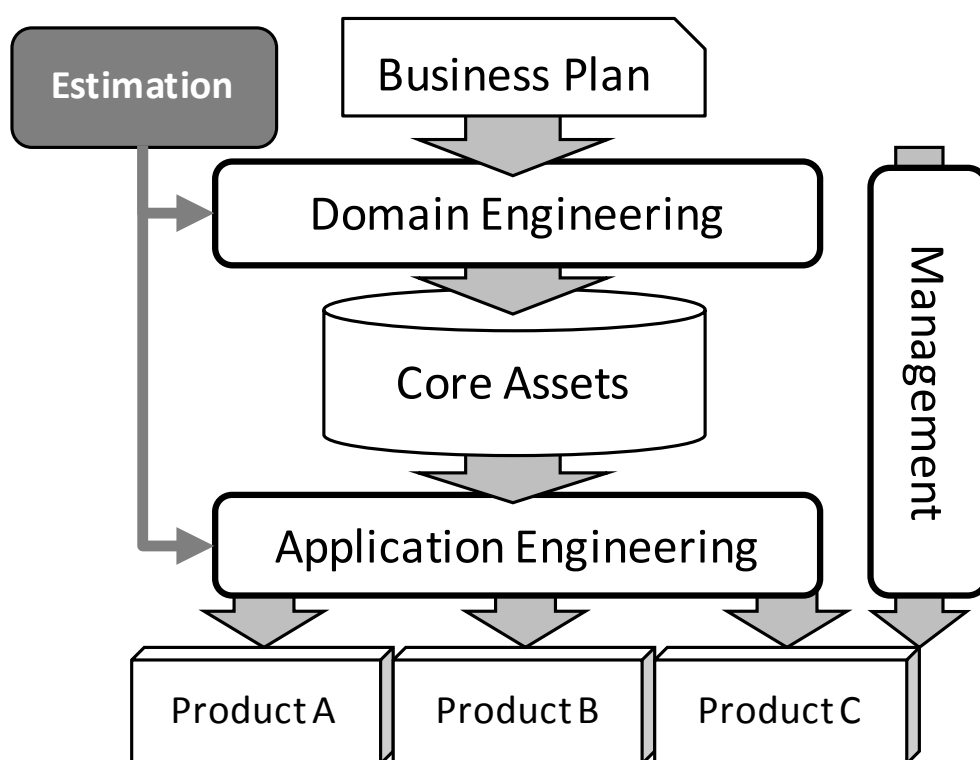



図 6-1　SPL の開発プロセス

SPL のアプローチには，製品とは独立にコア資産を開発する Proactive 型と一つの製品を雛型にしてコア資産を形成しながら次の製品を開発する Reactive 型がある[36]．本論文では少品種非量産系の製品を対象にしているため Reactive 型を採用している．少品種非量産系における Proactive 型の問題点を以下にあげる．

①コア資産の設計範囲があいまい：　長期開発で少品種非量産のためコア資産の基礎となるソースコードが少なく，また直近の開発以外はソースコードが古く参考になりにくい．

②必要以上の汎用性：　参考が少ないため必要とされる以上の汎用性を求める可能性がある．

以上の理由により，少品種非量産系の AIMS 開発においてコア資産は Reactive 型で構築される．

６.２　目標とするコア資産の姿

(1) 目標とするコア資産の姿

SPL の目的は, 製品シリーズにおいて開発すべき製品が N 個あった場合, 個別に N 個の製品開発を行うよりも開発期間，開発コスト，品質の面で優位な開発手法を提供することにある. そのためにコア資産を設け, N 個の製品開発負担を軽減する. では, そのようなコア資産はどのような形が求められるであろうか．以下，新たな分析装置を接続しても変わらないプラットフォームの部位がコア資産であるのに対して，分析装置固有に追加される部位をアプリケーションと呼ぶ.

ここで，A，B，C の 3 製品をコア資産無しで独立に開発することを考える．この場合 3 つの製品開発が独立に行われるので，本当は共有すべきツールや設計やコードなどが重複して作られることになる．このように個別に製品を開発するコストを個別開発コストと呼ぶことにする．次に，コア資産を形成して 3 製品を開発することを考える．するとコア資産によって重複を減少させることができる．コア資産をどんどん大きくしていくと極限において 3 製品一体のコア資産となる．3 製品のコードが共通一体となりパラメータを調整する範囲の構成管理で 3 製品が実現できるかもしれない．このパラメータの内，ユーザ視点のシステム特性がフィーチャ[37]である．この究極のコア資産最大化によっても新たなコストが発生する．3 製品の外部仕様の差が大きい場合，各アーキテクチャ層で参照されるパラメータの数及び論理的な組合せは膨大になる．例えば，AIMS では装置通信の通信手順の違い，特殊な検査項目の有無によるデータの有無，保守操作画面の違いなどが組み合わさって製品が作られ，組合せに応じて設計が複雑となる．コア資産の開発にかかるコストをコア資産コストと呼ぶことにする．

個別開発コストとコア資産コストの関係を図 ６-２に示す．3 製品のソフトウェアが共用されている割合(図中，共用度)がゼロから 1 に変化する間，個別開発コストは

減少するがコア資産コストが増える．個別開発コストとコア資産コストを加えると総開発コストになり，総開発コストはコア資産比率がゼロと１の間の値をとるどこかで最小となる．ここが目標とするコア資産の規模となる．つまり共用度を上げてもコア資産コストが大きいようでは共用のメリットを享受できないので，バランスの検討が必要になる．

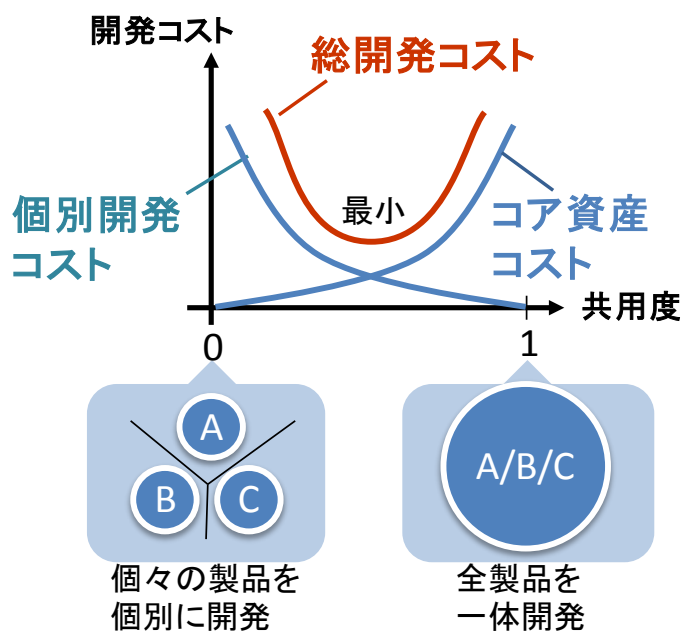



図　６‐２　製品 A, B, C 開発のコスト見積り

望まれる理想形を描くならば，コア資産と製品 A, B, C 固有のアプリケーションの関係は図　６‐３のようになる．コスト最小となるコア資産が製品 A, B, C のアプリケーションソフトと組み合わさって製品が開発される．このような理想の開発構成を見積り時点で統制する手法が求められる．

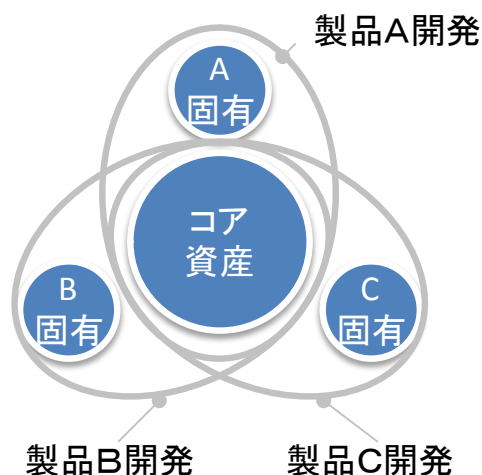



図　6‐3　コア資産と製品の関係

(2) AIMS におけるコア資産

上述の製品開発を AIMS に当てはめるため，製品 A, B, C をそれぞれ分析装置 a, b, c が接続された AIMS とする．上位互換性を求められるため，開発後の AIMS は，システム内に分析装置 a, b, c の全部または一部のいずれも接続可能にする必要がある．そのため AIMS はどの分析装置がどのような搬送経路で接続されているかをフィーチャとしてもつ．システム立ち上げ時に AIMS はシステムに接続される装置を認識して必要なソフトを立ち上げる（図　6‐4）．

装置構成に依存する部分が通信にとどまれば，これで統合は終了となる．しかし，前述したように分析装置の接続は AIMS のソフト全体に影響を与える．従って，コア資産とアプリケーションはシステム全体に散在する．この状況でコア資産を見積もれば，全体開発量の見積り精度が大きく揺らぎ，開発リスクが多大となる．全体開発量の見積り精度の向上が望まれる．

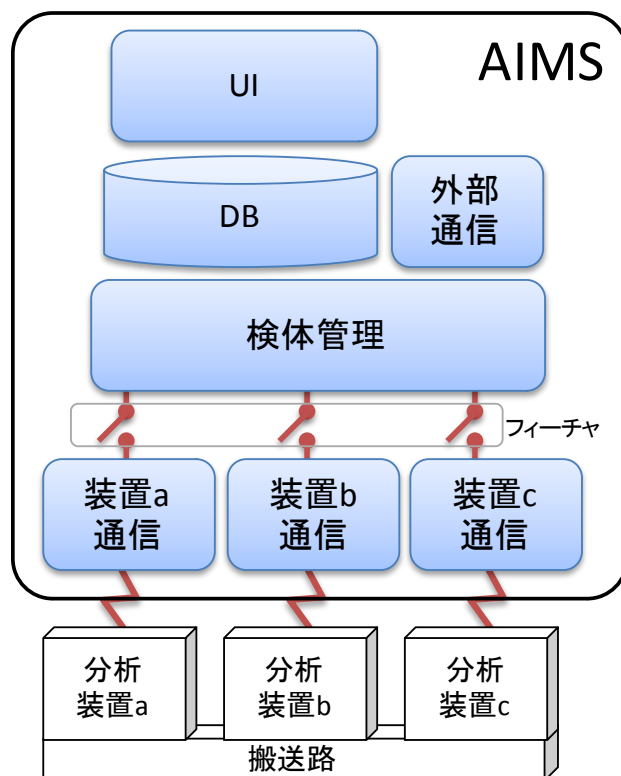



図　６-４　AIMS のアーキテクチャ

６.３　ADM

本論文ではコア資産を抽出する手法として Architecture Domain Matrix 手法(以下 ADM と呼ぶ)を提案する．ADM はドメイン知識要素，並びに統合システムのアーキテクチャ構成要素の 2 要素から成るマトリクスである(図　６-５).

ADM のセルに対応してソースコードを割り当て，セル単位にソースコードを分析し，コア資産にするか，または，分析装置固有の改造にするかを判定し，変更量の見積りを行う．ソースコードはドメイン要素とアーキテクチャ要素の対に位置づけられるため，業務知識から来る要求の範囲，機能実現の範囲が明確になる．

ADM 手法の狙いは，見積り時点でのコア資産コストを統制し，その見積り精度を向上させることにある．そこでこの 2 要素でソースコードの改造範囲を特定して見積り精度を向上させる．開発が予定された，要求仕様が確定している複数製品において，既存製品のソースコードを前述 2 要素で囲まれた範囲で解析し，共通なコア資産にす

べきか製品個別に開発すべきかを判定する．このような統制をマトリクスの全セルに対して行い，複数製品間を共通化するコストと個別に開発するコストのバランスを全体計画の中で図る．

ADM のアーキテクチャ要素の粒度は，一人のリーダーと開発メンバからなる機能チームで取りまとめられる範囲と規定する．この規定により，ADM のアーキテクチャ要素単位(図 ６-５における列単位)に変更を眺めると，どのようなスキルを持った人材がどのような変更を行うべきかが明らかになる．コア資産を含んでいる場合にはコア資産開発要員をアーキテクチャ要素に割りつける．すなわち，この粒度設定により ADM を使った開発チーム設計が容易になる．

また，ADM のドメイン要素の粒度はひとまとまりにシステムテストができる範囲と規定する．ADM のドメイン要素単位(図 ６-５における行単位)に変更を眺めると，ドメイン知識要素を実現するために必要な一連のアーキテクチャ要素変更が分かる．変更箇所と実現されるドメイン要素をまとめることで WBS が作成され，これにより変更仕様レビューの充実が図られる．加えて，どのアーキテクチャ要素の変更が最終テストに関連するかが明らかになる．すなわち，この粒度設定により ADM による WBS 導出が容易になりレビュー活動が円滑化される．

以上のようなコア資産開発要員の配置，及びコア資産開発者を含めたレビューの支援によりコア資産を含む開発計画の困難さを低減することができる．

本研究ではドメイン知識として業務フロー要素を用いている．ここで業務フローとは分析装置を立ち上げ，検査実行前の準備，検査の実行，保守，分析装置の終了処理まで，一日分の業務の流れである．報告[38]では医用装置のドメイン知識として業務フローを活用している．医用装置では医療業務フローを自動化することに注力しているため，業務フローがソースコード解析のための安定的なドメイン知識になる．また，業務フローは独立な業務工程から構成されている．アーキテクチャ要素はシステムの基本入出力に対応した独立な層構造になっている．アーキテクチャが独立な層構造であれば，ADM の２要素の独立性が確保でき安定したソースコード解析ツールとなる．ADM 手法は，長期間に渡り変更の可能性が少ないアーキテクチャ構成要素とドメイン知識要素が存在するような製品であり２要素が独立であることを前提としている．

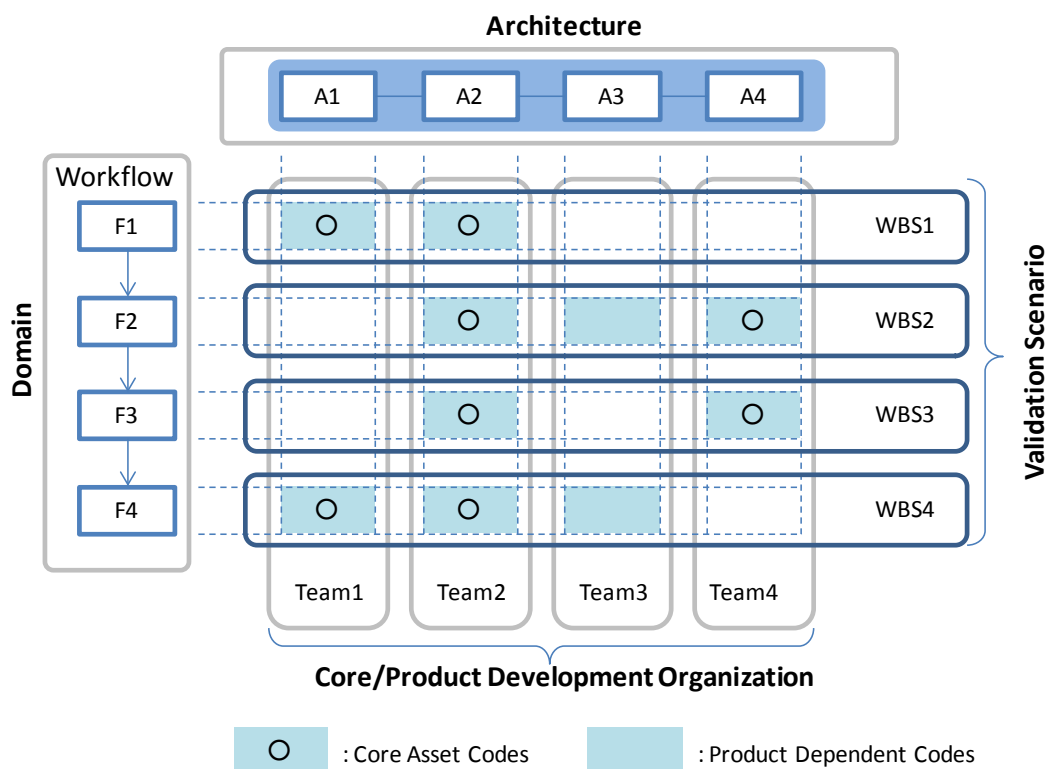



図　６-５　Architecture Domain Matrix

６.４　　開発コストのコントロール手順

図　６-６を用いて　ADM　手法の手順を説明する．入力は，過去に稼働実績のある AIMS(図中旧アーキテクチャ)と開発ロードマップである．開発ロードマップにはこれから接続される N 機種と出荷時期が具体的に記述される．

N 機種それぞれについて行列セルのソースコードを分析して，　N 機種個別にプログラム作成すべきか，それとも N 機種共通のコア資産とすべきかを検討し ADM 手法を実行する．

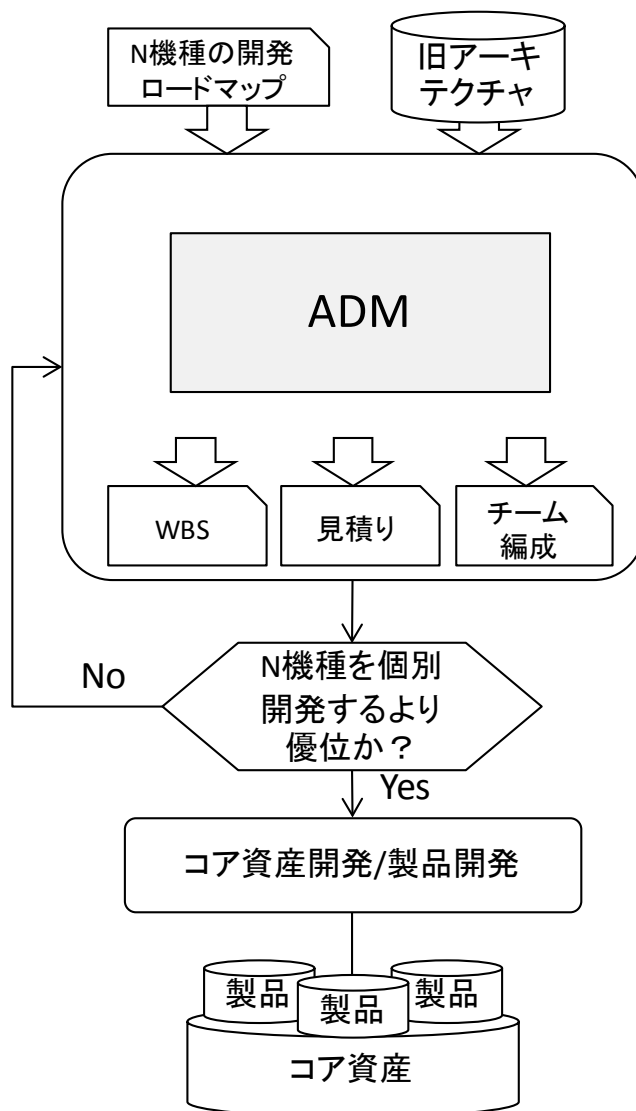



図　６-６　ADM 手法による開発の流れ

以上の検討結果として，コスト及び開発期間が N 機種を個別開発するより優位かを判定し優位であれば開発に着手する．優位でなければ，更に開発量を軽減する策を検討する．策は共通範囲(コア資産)を括りだすこと，または，既存の製品プログラムを機能重複覚悟で流用することの選択が考えられる．これにより既に述べた個別開発コストとコア資産コストとの調和を図る(図　６-２)．

ドメイン要素 i，アーキテクチャ要素 j における改造コスト $C_{i,j}$ は以下の式となる．

$$C_{i,j} = \min\left(CC_{i,j}, CD_{i,j}, CA_{i,j}\right) \qquad (6.4.1)$$

ここで，$CC_{i,j}$はコア資産に仕上げるコスト，$CD_{i,j}$はコア資産とアプリケーションを分離するコスト，$CA_{i,j}$は純粋アプリケーションを開発するコストである．3 つのコストのいずれが低いかは一義的に決まらない．開発体制にも依存し，どれくらい深くソースコードを解析するかにも依存する．コア資産に仕上げるためには複数の分析装置で共用できる確証が必要であり，それはアーキテクトの意見あるいはソースコードの見通しの良さにかかっている．ADM 手法では，アーキテクチャとドメインにより切り出された部位が他と独立であることを前提としているため $CC_{i,j}$ の積上げにより全体のコスト統制が図られる．

開発に入る前に，コア資産と決定されたソースコードはコア資産を取り扱う責任者のみが改変できるようパスワード付の構成管理下に置かれる．このコア資産を変更するに当たっては，一部の製品だけが開発量減の利益を得て他の製品が法外な開発量増の不利益となることのないようコア資産チームと製品チームとの共同レビューを行って決定する．

以下に ADM 手法の具体的な手順を述べる．

(Step 1) 開発計画の設定

実際に稼動する品質の確定したシステムを選定する．これは図 6-6 の旧アーキテクチャに相当しベースプロダクトと呼ぶ．加えて並行開発する N 機種の開発計画を確定させる．ベースプロダクトとの相似性から N 機種は絞られる．

(Step 2) ベースプロダクトの解析

図 6-5 に相当する表をベースプロダクトについて作成する．まず，ベースプロダクトのソースコードを分類して，アーキテクチャ要素とドメイン要素の行列セルに対応させる．次にベースプロダクトソースコードを解析して，行列セル毎にソースコードを①コア資産として今後共通に使われるプログラム，②コア資産を含むプログラムでコア資産としてソースコードを分離されるべきプログラム，③アプリケーションと

して製品個々に任されるプログラム，の 3 種類に分類する．便法として①,②,③を c (core), d (dual), a (application)と記号化し表内に記す．②,③が N 機種の可変部位を吸収する変更部位になる．

(Step 3) N 機種の見積り

図 6‐5 に相当する表を N 機種について作成する．N 個の表の中身(c,d,a)はベースプロダクトの表と同じであるが，c,d,a の 3 種類に分類されたそれぞれのソースコードにおいて，改造・追加が必要なものにはプライム(’)を付ける．プライムのついたコードの意味は以下の通りである．

①c’：コア資産として共通に使われるために改造する．

②d’：コア資産として共通に使われる要素を含んでいるため，その共通要素を抽出するために改造する．

③a’：個別機種対応のソースコードではあるが，他のコア資産と接続するために改造する．

ベースプロダクトの表にもプライムが付く可能性がある．このプライムは，ベースプロダクト内のコア資産が N 製品を吸収するにあたり変更が発生するということを意味する．表内には c,d,a,c',d',a'が記され，この内 c 以外は改造・追加になるので，これらから N 機種の全体開発工数を見積もる．この時点で c のコア資産が少なすぎると，共通化の効果が薄く全体開発工数は大きくなる．一方，c',d',a'が多すぎるとコア資産の存在によって返って工数が増えることになる．c を大きくするか c',d',a'を小さくするかの可能性がある場合には(Step 2)に戻り，コア資産を再計画する．コア資産の増減が全体開発工数の削減に寄与しなくなったならば，最終結果として全体開発工数を受け入れる．

(Step 4) チーム編成設計

チームはアーキテクチャ要素単位に形成される．ADM の表を縦方向(アーキテクチャ要素単位)にたどって開発工数を加算し，その変更量と難易度に応じて現状人材のスキルを照らし合わせ人材を配置する．アーキテクチャ要素内にコア資産が存在する場合コア資産開発者を割り当てる．

(Step 5) WBS 設計

ADM を行単位に眺め，業務フロー要素ごとに変更点を集める．これらを業務フロー要素の実現のための改造一覧，つまり WBS にまとめ以下のように活用する．

①アーキテクチャ要素の検証が終わった後，システム全体として妥当性をテストするときの拠り所とする．

②アーキテクチャ要素間のインタフェースを明確にして各アーキテクチャ要素を変更するチーム間でレビューをガイドする．

６.５　まとめ

本章では，コア資産開発を見積り時点でコントロールする開発手法として ADM 手法を提案した．

6.1 節では SPL におけるエンジニアリング活動において，ドメインエンジニアリングとアプリケーションエンジニアリングを支援する見積り活動に本論文の開発手法を位置付けている．また，SPL には大きく分類して Proactive 型手法と Reactive 型手法があるが, 本手法は非量産であることを理由に Reactive 型手法を選択していることを示した．

6.2 節では目標とするコア資産がどのようなバランスの中で位置付けられるかを論じた. 複数製品を構築するとき全く別個に開発することによる重複開発コスト, また, 全てを一体として開発して部品の組合せで製品を実現することによる部品管理コストの両極端の状況を説明し，本論文はその中間点でコストバランスすることを目指している．従って，一方的にコア資産を増加させるのでもなければ，一方的にアプリケーションを増加させるのでもない．両者のバランスをとる見積りが求められる．AIMS におけるプラットフォームは分析装置を追加することにより通信部位が変更となる．しかし，その変更範囲は第 2 章で述べたように拡大され，アーキテクチャ内の随所に渡る．従って，バランスされるコア資産とアプリケーションは共に AIMS 内に散在する．コア資産とアプリケーションのバランスをとる見積りにおいては，ソースコード全体に渡る解析が必要となることが示された．

6.3 節では ADM 手法を提案した．この手法では，アーキテクチャ要素とドメイン要素(本論文では装置の操作フローであるワークフロー要素を採用)の 2 軸からなる表にソースコードを割り当て，表のセル単位にソースコードを分析してコア資産にするかアプリケーションにするかを決定する．従来であればソースコードのリファクタリングはアーキテクチャ要素単位にしか行われないが，本手法は見積り精度を向上させるためにドメイン要素を加えて見積りの分解能を上げている．アーキテクチャ要素が機能，ドメイン要素が要求に相当するため，表のセルは機能と要求の対により囲まれていることになり，独立な仕様検討範囲を提供する．また，この表を横に眺めることにより変更仕様レビューに活用できる WBS を作成でき，縦に眺めることにより変更に対する開発要員の配置に対して有益な情報を与える．

6.4 節で ADM 手法の手順をステップ毎に定義した．コア資産を抽出する元となるベースプロダクトと接続すべき N 機種の仕様を選択し，ベースプロダクト及び N 機種について ADM の表を作成する．表により見積りを行い，コア資産が少なすぎる，あるいは，変更が多すぎる場合には再度見積りを検討するという繰り返しのステップを含んでいる．これによりコア資産とアプリケーションのコストバランスを取る．

以上述べたように ADM 手法は見積りのための表と見積りの手順から構成され，これにより目標としているコア資産とアプリケーションの調和を見積り時点で達成する．表からは，変更管理レビュー及び開発者配置に関する情報を抽出することもできる．以上によりコア資産の統制, 見積り精度向上, 長期開発の支援という課題を解決する．

第７章　ADM 手法の適用と結果評価

本章では前章で導入した ADM 手法を実際のプロジェクトに適用した結果とその評価を述べる．

7.1 節では，対象となったプロジェクトの計画について述べる．

7.2 節では ADM 手法を当該プロジェクトの見積りに適用した見積り結果を述べる．7.2.1 節はコア資産を抽出する元となるベースプロダクトと，プロジェクトで計画された３機種において，ADM 手法で行ったコア資産分析結果を示す．7.2.2 節では ADM 手法で得られた見積り値，7.2.3 節ではチーム編成設計，7.2.4 節では WBS 設計の結果を述べる．

7.3 節では ADM 手法を適用して見積もられたプロジェクトがその後開発を終えた時点で開発を振り返り，成果を見積りと比較して評価する．

7.4 節では，ADM 手法特有の効果について考察する．

7.5 節では本章を結論付ける．

７.１　対象プロジェクト

ADM 手法を適用した実プロジェクトについて述べる．事業戦略から与えられた開発課題は新たな 3 機種の分析装置を 1.5 年で結合するものであった．具体的な月単位の開発ロードマップを表 7‐1 に記す．初年度 3 月までに完了する製品をベースプロダクトとした．網掛け部位が開発工程である．

表 7‐1　ADM 手法適用対象プロジェクトの計画

	初年度									次年度								
	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9
ベースプロダクト																		
分析装置A																		
分析装置B																		
分析装置C																		

分析装置 A, B の 2 本の開発は，それぞれ設計に始まり実装・テスト・QA 完了までの工程となり，C は QA を含まない装置性能検証の工程である．ここで，QA とは組込みソフトウェアの検査であり，これを終えるとシステムの妥当性を検証する網羅的なシステムバリデーションが社内及び社外で実施される．このシステムバリデーションが完了して初めて医療機関に製品としてシステムが提供される．B のスタートが遅れているのは分析装置ハードウェアの開発と同期させた結果である．

７.２　ADM 手法適用結果

７.２.１ ベースプロダクトの解析と 3 機種の見積り

表 7‐2 にベースプロダクトと分析装置 A, B, C 3 機種を加えた ADM の分析最終結果を示す．行には AIMS の業務フロー要素の大分類と中分類を並べ，列にはアーキテクチャ要素を並べている．最下段に分析装置の区分けが記されている．P はベースプロダクトを示す．表内にはソースコードを分類して前述の(Step 3)で説明したプライム付記号 c',d',a'を付けている．

表 7-2　P, A, B, C 製品の ADM

	ドメイン		アーキテクチャ																							
	大分類	中分類	UI				DB				検体管理				装置通信				分析装置				外部通信			
1	STARTUP(SU)		c’	c	c	c	d	d	d	d’	c’	c	c	c	d	d	d	d’	a	a	a	a				
2	全体管理(GM)	GM1	d	d	d	d	d	d	d	d	a	a	a	a	d	d	d	d	a	a	a	a				
3		GM2	d’	d’	d’	d	d	d	d	d	a	a	a	a	d	d	d	d	a	a	a	a				
4		GM3	d’				d	d	d	d	a’	a	a	a	d	d	d	d	a	a	a	a				
5		GM4	d	d	d	d																				
6	試薬準備(RP)	RP1	d	d’	d’	d’	d	d	d’	d’	a	a	a’	a	d	d	d	d	a	a	a	a				
7		RP2	d	d	d	d	d	d	d	d	a	a	a	a	d	d	d	d	a	a	a	a				
8		RP3	d	d	d	d’	d’	d	d	d’	a’	a	a	a’	d	d	d	d’	a’	a	a	a’				
9		RP4	d	d	d	d	d	d	d’	d	a	a	a’	a	d	d	d	d	a	a	a	a				
10	測定(M)	M1					d	d’	d’	d’	c	c’	c’	c’	d	d’	d’	d’	a	a’	a’	a’				
11		M2					d	d’	d’	d’	c	c’	c’	c’	d	d’	d’	d’	a	a’	a’	a’				
12		M3					d’	d’	d’	d’	c’	c’	c’	c’	d	d’	d’	d’	a’	a’	a’	a’				
13		M4	d	d’	d’	d’	d	d	d’	d’																
14	キャリブレーション(C)	C1	d	d	d	d’	d	d	d	d’	c	c	c	c’									c	c	c	c
15		C2	d’	d	d’	d’	d’	d	d’	d	c’	c	c’	c’									c	c	c	c
16	精度管理(QC)	QC1	d	d	d	d’	d	d	d	d’	c	c	c	c’									c	c	c	c
17		QC2	d	d	d’	d					c	c	c’	c’									c	c	c’	c’
18		QC3	d	d	d	d																	c	c	c	c
19	検査業務(I)	I1	d’	d	d	d	d’	d	d	d													c	c	c	c
20		I2	d	d	d	d	d	d	d	d													c	c	c	c
21		I3	d’	d	d	d’	d	d	d	d’	c	c	c’	c’									c	c	c	c
22		I4	d’	d’	d	d’	d	d	d’	d	c	c	c’	c									c	c	c	c
23		I5	d	d	d	d’					c	c	c	c	d	d	d	d	a	a	a	a’				
24		I6									c	c	c	c	d	d	d	d	a	a	a	a				
25	SHUTDOWN(SD)		c	c	c	c	c	c	c	c	c	c	c	c	d	d	d	d	a	a	a	a				
		分析装置	P	A	B	C	P	A	B	C	P	A	B	C	P	A	B	C	P	A	B	C	P	A	B	C

(Step 2)から(Step 3)の分析の途上，コア資産を積極的に形成する方向に統制した例として検体管理，その逆に消極的に統制した例として装置通信があった．検体管理はシステムをモデリングする部位であり抽象化が可能と判断しコア資産の幅を広げた．装置通信では分析装置 A, B, C の通信形式の違いが大きく，違いを全て吸収する構成容易なコア資産を作るには工数が大き過ぎるため最低限のコア資産形成とした．

最終的に, 表中行数に当たる 25 のドメイン領域中変更は発生しないもの（例えばドメインの No.2 や 5 など)と変更が発生するものが現れている. 変更を示すプライムは, ベースプロダクト内のコア資産が 3 製品を吸収するにあたり変更が発生するということを意味する．事業計画における現実的な変更要求である 3 機種を頼りに，これらが最適に接続できるために再利用性をいかに高められるかを追求している．

7.2.2 見積り結果

表　7‑3 にベースプロダクトに要した開発工数を 100 としたときの本プロジェクト見積りを示す.

表　7‑3　開発コスト見積り

ベースプロダクト	本プロジェクト				
	コア資産	A接続	B接続	C接続	計
100	11.3	18.9	34.0	52.8	117.0

ベースプロダクトの開発実績に比べて, 見積りでは 17%増しで 3 機種開発の見通しを得た．前述したように ADM 手法では個別開発コストとコア資産コストとの調和を図るという設計行為が含まれているため，17%増しに収める設計ができたといえる．ここで，C 接続の開発工数が他に比べて大きいが，これは分析装置自体に開発要素があったためである．見積りは本プロジェクトの開始可否を判定する試金石であり，この結果は本プロジェクトの成功率を高めた．また，業務フロー要素という要求仕様とアーキテクチャ要素という実現機能が明確になりソースコードの自由度が限定されたことも見積り精度を向上させている.

7.2.3 チーム編成設計

表　7‑4　チーム編成設計の結果概要

	UI	DB	検体管理	装置通信	分析装置	外部通信
コア資産開発者	○	○	○	○		○
分析装置単位にチーム編成	○		○		○	

表　7‑4 にチーム編成設計の結果概要を示す．アーキテクチャ要素単位にコア資産開発者を割り当てるか，分析装置単位にチーム編成をするかを検討した(表中の○).

コア資産の開発には各アーキテクチャ要素で技術的にリードできる人材を最低一名あてた．

DB，装置通信，外部通信については開発量が比較的少ないこととその標準的な性格から，分析装置単位のチーム編成は行わずコア資産開発者のみを配置した．分析装置は全てコア資産を含まない分析装置依存のコードであるため，分析装置単位のチーム編成とした．更に，アーキテクチャ要素毎に必要となる人員を見積り割り当てる．アーキテクチャ要素毎にコア資産とアプリケーションの開発量が分かるので，これをコード生産性（LOC/工数）(LOC: Line Of Code)で除することによりそれぞれの必要人員が算出される．具体的なチーム編成にあたっては，本来横割り組織であるコア資産チーム及び，限られたサブリーダをいかに配置するかが重要となる．これについては 7.4 節の(5)コア資産チームの所属，(6)サブリーダの選出で述べる．

このようにマトリクス状に組織配置を検討することにより，技術的な難易度と具体的な人のスキルとを調和させることが可能となり，開発のリスク軽減が図られた．

7.2.4 WBS 設計

図　7-1 は本プロジェクトで作成した WBS 仕様書の一例を簡略化して示している．要件名称及び内容に，業務フロー要素が書かれ，下段のアーキテクチャ要素ごとに変更内容が書かれている．この例では保守ツールの実行のために，画面，DB，検体管理，装置通信，分析装置の変更が連携することが記される．この WBS を元に，アーキテクチャ要素担当者は設計書を書き，ここに上がっている関係者が集まりレビューを行い実装に入る．各要素は設計書に基づき試験され，後に組み合わせた時点で，システムテスト「装置 C 固有の保守ツールを実行する」によって最終テストを終える．

変更点レビューは，ソフトウェアの品質を早期に高めるためには重要な工程である．これを怠れば，後工程であるテスト工程で不具合が発生しその原因探索を含めると多大なコストの発生，開発遅延を招く．変更点レビューの中でもアーキテクチャ要素間のインタフェース確認は重要であり，あるアーキテクチャ要素の技術者は他の要素の技術者とレビューして間違えのないインタフェースを確立する必要がある．そのとき

この WBS は有効なレビューツールとなる．すなわち，最終システムテスト仕様となるドメイン要素に対してどのアーキテクチャ要素の変更が関わっているかが明確になる．WBS を用いることでレビュー会議での招集者及び議論すべきインタフェースが明確になり品質向上に貢献できた．

このように，ADM 手法では WBS を導出できるため，アーキテクチャにおける全体と部分の掌握が容易であり，最終試験のあり方が明示されるため，要素開発を完結することができ，開発効率向上への貢献が大きい．

No.	要件名称	要件内容
	装置C用保守ツール	装置C固有の保守ツールを実行する。

関連プロジェクト				要求元	優先度	採否	採否コメント
P	A	B	C				
			○				

アーキテクチャ要素	詳細要素	新規・改造内容	改造量
外部通信			
UI	System	パラメータ設定及び実行の画面を追加。	
	Overview	保守の実行状況の表示。	
DB		保守パラメータ登録。	
検体管理	制御	保守に使用する検体制御。	
	状態処理	保守の状態モニタリング。	
装置通信		保守の指令とパラメータ転送。	
分析装置	保守管理	保守の実行制御と実行状況報告。	

図 7-1　ADM 手法から導出される WBS 例

7.3　ADM 手法で見積もったプロジェクトの結果と考察

ADM 手法を適用して得られた以上のような見積りや計画に基づき開発が行われた．

その結果を表　7-5 に示す.

実際には, 本プロジェクトに先立つベースプロダクトの開発が 2 ヶ月遅延したため, 本プロジェクトの開始は 2 ヶ月遅延した. しかし, プロジェクト A においては 3 ヶ月, プロジェクト B においては 2 ヶ月前倒しにて完了することができた. プロジェクト C については分析装置の機能追加が行われたため 2 ヶ月の遅延となったものの，全体として，1.5 年で 3 機種の開発を完了させ計画に適合したコスト統制ができた．過去の自社内実績では SPL を使わない開発で 2 機種開発に 2.5 年をかけている. この 2 機種開発ともにその規模は,本論文における分析装置 A,B,C を平均した規模に比してほぼ対等といえる. 従って, 年当りの開発機種数は 0.8(機種/年)から 2.0(機種/年)と改善し, 2.5 倍の生産性を得たといえる.

表　7-5　プロジェクトの実績進度

	初年度									次年度										
	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11
ベースプロダクト																				
分析装置A																				
分析装置B																				
分析装置C																				

開発のゴールを達成することができた理由として，見積り精度が向上したこと，及び，再利用コードが増えて改造量を抑えられたことがあげられる．3 機種を開発完了するための見積り工数は表　7-3 により，ベースプロダクトの開発工数に比べ 1.17 倍であった．ベースプロダクト開発は SPL を用いない 1 機種開発であったため，その工数は 2.5(年)/2(機種)×(全人員)=1.25(年・全人員)となる．これを 1.17 倍した 1.46(年・全人員)が 3 機種開発の見積り工数である．これに対して実績工数は，ベースプロダクト開発と本プロジェクトとの間で開発人員に差を設けていないことを考慮すると，1.5(年・全人員)であった．見積り誤差は+2.7%であり実用的な見積り値が得られたといえる．

再利用コードについては，コア資産の中核を成す UI・DB・検体管理の総ソースコード行数におけるコア資産の割合を表　7-6 に示す．コア資産以外の行数は分析装置

固有のソースコードとなる．見積りに対して実績は A, B の接続において差が少なく，全体開発を揺らぎなく行う程度に見積り精度は高い．また，A, B の接続では見積り以上のコア資産率を実績で出しており，開発の前倒しに貢献している．これはベースプロダクトの改造の中でより高く均衡ある共用度が実現できたためである．一方，分析装置 C では見積りに比べてコア資産率の実績が下がっている．これは分析装置固有の機能追加によるためで，実際進捗の遅れが出ている．性能検証のための装置には仕様の揺らぎがあり実際見積り以上の期間がかかっている．しかし，分析装置 A, B の開発前倒しがこの遅れをカバーする形となり，また，60%以上のコア資産率はコア資産の有用性を示している．

表 7-6　UI・DB・検体管理におけるコア資産の割合

	A接続	B接続	C接続
見積り	95.4	93.7	76.0
実績	96.5	98.1	60.8

また，ADM 手法で生成された WBS が開発の中で有効に活用された．特にアーキテクチャ単位に編成されたチームはそれぞれの専門領域において独立性を維持しながら，WBS を通じてレビュー，システムテストで全体の整合性をチェックすることができた．この点でコア資産を含む開発計画の困難さを低減できたといえる．

なお，ここで抽出されたコア資産は，その後 4 年間で新分析装置 5 機種，新搬送路 2 機種接続に活用されている．これは抽出されたコア資産が実効的であったことを示す．

ADM 手法は，長期間に渡り変更の可能性が少ないアーキテクチャ構成要素とドメイン知識要素が存在するような製品開発を前提にしている．医用機器向けの規制により業務が規律されている医用製品においては比較的安定した業務フローがドメイン知識要素として利用できる．従って，規制適用が求められる製品においてはアーキテクチャ構成要素が安定していれば，ADM 手法の適用可能性は高く，本事例はそれを例

示している．

７.４　ADM 手法特有の効果に対する考察

一年以上のプロジェクトを通して，コア資産として参照する材料が少ない長期間の開発において生産性を向上させる手法として ADM 手法を評価してきた．この手法の特徴を以下まとめる．

(1) コア資産開発とアプリケーション開発の調和

SPL には reactive と proactive の両アプローチがあるが, 開発では reactive アプローチを採用している．proactive アプローチでコア資産を製品開発と切り離して開発するにはある程度のコア資産を生み出す材料が必要であり，それを十分には用意できなかったことが reactive 採用の主な理由である．それに加えて，長期開発であるためにコア資産開発の費用回収問題がある．

proactive では，コア資産を集中して開発し，後続で開発される製品群にできるだけ多くコア資産を活用する．コア資産の開発コストは，その製品群によって賄われ，言わば投資回収が行われる．より多く活用されることで，コストが回収され，多くの製品を生み出し，顧客の満足を得る．一方，reactive かつ長期開発の場合，同様に，コア資産が長期開発の中でできるだけ多くの顧客満足を得ることが求められる．従って，この一回の開発内では過剰なコア資産を作らないよう注意を要する．ADM 手法で考える開発のリスクは，余りに汎用性を追いすぎて開発チームのスキルを超えているかもしれないコア資産を開発しようとして，製品の開発までもが遅延してしまうかもしれないことである．コア資産の共用性は計画された新たな製品の仕様群の差を比較検討することにより抽出される．仕様の差が大きすぎて 1 ソースコードで実現することが難しくなると，計画よりはかなりの開発期間超過が想定される．例えば，ADM 手法の適用事例では，アーキテクチャ要素の装置通信におけるコア資産を低減した．通信ソフトは一般的に標準化が可能な領域であり，通信階層，パラメータ，ライブラリなどにより製品仕様の差を吸収することができる．しかし，適用事例では仕様差を吸収するためにコア資産を作らず，製品毎のアプリケーションソフトで吸収すること

に決めた．理由は，通信仕様の差を解析した結果，余りに多きな仕様の開きを見つけたためである．

コア資産を低減した例を述べたが，一方でコア資産を増加させる例もあった．検体管理におけるコア資産を見積もったとき，仕様差はかなりあったものの，その大抵をコア資産で実装することを決めた．これはソフトウェアの抽象度を高めて仕様差を吸収することになった．理由は，この検体管理こそが AIMS の根幹であり重要なコア資産になりうると判断したからである．

このように ADM 手法によって鳥瞰的な見方ができた．加えて ADM 手法により，コア資産とアプリケーションとの調和を図ることができた．コア資産とアプリケーションとの関係は ADM 手法の各ステップで確認される．まず，両者はアーキテクチャ要素内で混在し，ADM 表におけるドメイン要素により分解される．次に，ADM 表のセルにおいて分析が行われ，いくつかのセルではコア資産，アプリケーション，または，両者に改造を加える判断をする．これはコア資産とアプリケーションは協調して共存しなければならないことを意味する．つまり，「コア資産を作ったから，これを使うべし」でも「コア資産が必要だから，作るべし」でもない．どちらも改造対象であり，それを決定するのは開発全体のコストである．このようなコスト調整の場を ADM 手法は提供している．

⑵ コア資産としての ADM 表

ADM 手法適用事例において可変性のスコープを開発ロードマップに計画された 3 製品に限定した．この決定は仕様コントロールの巾を明確にするために行われた．ここで，3 製品だけを適用対象としたコア資産は，本当の意味のコア資産，つまり，できるだけ多くの製品を適用対象にできる共用性の高い資産には成りえないと考える向きもあるであろう．しかし，コア資産はソフトウェアに限られない．3 製品の仕様差を検討して作られた ADM 表もコア資産となる．この表は既存製品と将来製品との違いを示しており，次の開発の機会でこの表を仕様変更管理ツールに使うことができる．例えば，新たに別の装置を接続したい場合である．ADM 表には既に検討されたコア

資産がソースコード内で位置付けられており，その知見によってかなりの詳細レベルまでソースコードの変更計画を見通すことができる．そして，ADM 表に関わった技術者にも，仕様変更に対するソースコードの変更管理方法が身に付いている．本研究で扱ったコア資産は後に新たな 5 分析装置，新たな 2 搬送ラインに接続されており，これはコア資産の耐久性を説明している．

(3) ADM 手法見積りの準備

ADM 手法はソースコードをアーキテクチャ・ドメインの刻みに分割してコア資産とアプリケーションの設計を見積もる手法である．これによって結果として見積り精度が向上した．このプロジェクトを始める前に，開発チームにおいてドメイン知識をより深く理解するための活動を行っている．開発に先立って 2 年前に SPL を専門とする研究者に参画してもらい，製品に用いられるドメイン知識の解析を行っている．特に医療機器という特殊性もあり，専門性が高く，誰もが分かる知識とは言えなかったためである．ドメイン知識はソースコードとの関係においてまとめられ，各アーキテクチャ要素のサブリーダに共有された．これはアーキテクチャとドメインの両面でソースコードを分析する ADM 手法にとってかなりの助けになった．そして，開発者全員が ADM 手法の見積りを信用することにつながった．

(4) 規制が求められる業界における活用

適用事例において目的が達成された理由は，改造量が統制され，見積りの精度が向上し，ソースコードの再利用が高まるという流れから生まれたと考えられる．アーキテクチャ要素とドメイン要素から成る ADM 表がこの流れに貢献している．しかし，ADM 表がどのようなタイプのプロジェクトにでも適用できるとは考えていない．ADM 手法の適用事例から推定すると，比較的大規模で長期のプロジェクトで，かつ，どのドメイン知識もかなり専門的で安定している場合に ADM 手法が有効になる場面があると考える．その点で，1 つの可能性として ADM 手法は規制が求められる業界において有効性が期待できる．理由は，そのような業界では規制への適用が求められ

る余り開発プロセスの厳格性が求められ長期開発になりがちだからである．規制はドメイン世界における業務プロセスを規定し，顧客視点においてもかなり安定した業務が設定される．

例えば，計測に関わる研究所や検査機関は，SOP(Standard Operating Procedures)を用意することが一般的に求められている．全業務者はこの SOP に従い業務を行う．SOP には検査機器の校正管理や品質管理の手順が書かれている．どんなに特殊な装置を導入したとしても，業務者はデータ品質を管理するために SOP を守る．従って，検査機器の仕様は自然と一般的な SOP に適用するはずである．機器が装置制御だけでなくデータオートメーションにも関連するならば安定的な SOP を実装することが求められ，機器がもつべき業務関連のドメイン知識を安定化させる．

(5) コア資産チームの所属

適用事例であげたベースプロダクトの開発には，経験豊富で優秀な技術者が参画した．この技術者はコア資産の開発チームメンバの候補者となった．同時に，アプリケーション開発チームにおいても，この技術者は活躍が期待されていた．一方，同じ人を両チームに跨って配置することは，能力的に可能としても，開発原理からは難しい．コア資産が一部のアプリケーションの利益に傾くことが無いように，コア資産とアプリケーションは独立に開発すべきだからである．そこで，この優秀な技術者の知見が開発全体で共有されるように，妥協的なチーム編成を採用した．すなわち，その技術者の多くをコア資産チームに割り当てて，横断的な要員としてアプリケーション開発チームに分散配置させた[39]．分散配置された技術者は独立性を維持しながらコア資産の開発を行うが，同時にアプリケーション開発の進捗状況を共有する．また必要に応じてコア資産とのインタフェースを議論する．開発は独立に，しかし，コミュニケーションは円滑に行える配置として適切だったと評価している．

(6) サブリーダの選出

優秀な技術者は有効に開発チームに配置されるべきであり，チームのサブリーダの

選出もその技術者から配置されるべきである．ADM 手法はその選出を助けるツールにもなった．サブリーダ候補は例えば画面といった特定のアーキテクチャ要素の専門化である．限られた人数の技術者を効率的な管理を目指してコア資産とアプリケーションの開発に配置するには，一定の論理付けが必要となる．そこで，ADM 表から各アーキテクチャ要素のソースコード行数が占める全ソースコードにおける割合を算出し，その割合をサブリーダ候補者の全人数に掛けて，各アーキテクチャ要素に割り当てる人数見積りを算出した(図　7‐2)．この見積り値は一般に小数点を含むので，対象プログラムの難易度を勘案しながら，整数化する．もし，複数の要素に渡って見積り値が１に満たない場合には，一人のサブリーダを横断的に配置した．

サブリーダの配置は難しい管理業務であり，配置されたサブリーダの不満が開発全体を揺るがしかねない．このような不満が助長する原因の１つは，サブリーダ全員におけるコンセンサスの欠如にある．配置理由について客観的に考える上で，上述の簡単な計算は大きな役割を果たし，これをサブリーダ全員に公開して議論することで，開発全体を理解した配置を納得してもらう機会を得た．

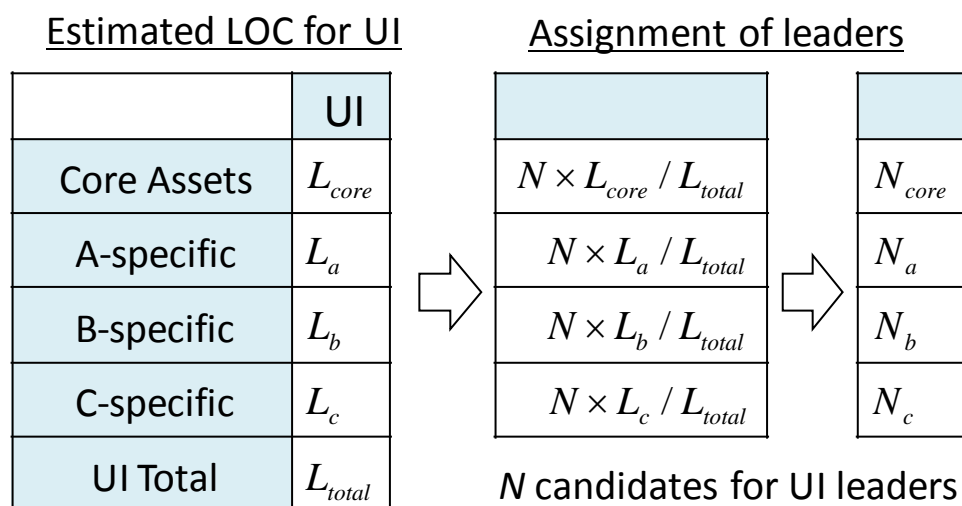



図　7‐2　サブリーダ選出のための計算

(7) Work Breakdown Structure (WBS)

WBS が開発全体に有効に作用したことは適用事例の考察で述べた．組織的にはコミュニケーションツールとしての役割が大きかった．技術者は大きくアーキテクチャ要素向け技術者，分析装置向け技術者，コア資産開発向け技術者に分かれる．WBS はこれらの技術者の間のコミュニケーションを円滑化した．特にこれら技術者がソースコードを改造するにあたって重要なことは，自分と異なる分野のソフトウェアとのインタフェースである．自分が行う変更が最終的にどのようなドメイン要素を支えることになるのかについて感度が高まり，従って，その関連する変更点の開発者とインタフェースを議論することを促進した．開発領域の違いを越えてコミュニケーションするツールとして機能し，結果として生産性が向上した．

7.5　まとめ

本章では前章で導入した ADM 手法を 1.5 年の実プロジェクトに適用し，見積もった結果を基に開発を行った．

7.1 節では，対象となったプロジェクトの計画を設定した．3 機種の分析装置を 1.5 年で結合する計画であった．

7.2 節では ADM 手法を当該プロジェクトの見積りに適用した見積り結果を述べた．7.2.1 節は ADM 手法の見積り結果として ADM 表を一覧した．コア資産はアーキテクチャ各所に存在しており，そのようなコア資産の摘み取りが ADM によって可能になったことが分かる．手順の**(Step 2)**から**(Step 3)**の途上では，少ないコア資産を増やす例，及び，変更量が多すぎるためにコア資産を減らした例を述べた．このような手順の中で最適なコア資産が形成されていった．7.2.2 節では見積り結果が，ベースプロダクトに要した工数を 100 とすれば本プロジェクトは 117 であった．純粋にコア資産無しで開発するならば 3 機種を開発するためには 300 の工数を要するところ，117 であったことは十分なコア資産を抽出できたといえる．7.2.3 節ではチーム編成設計において ADM を縦に眺めることによりアーキテクチャ要素毎にどの程度コア資産チームを配置すべきかが示された．このような配慮が開発のリスク軽減に貢献している．7.2.4 節では ADM から導かれた WBS の例を示した．どのような変更がドメイン要素であるワークフローに関わるかが WBS を通じてレビューされる．

7.3 節では以上の見積りを終えたプロジェクトが 1 年以上をかけて開発を完了した時点での結果を考察した．プロジェクトは本開発とは別の理由で開始が 2 カ月遅れたが，着手から完了までの期間は見積り通り 1.5 年であった．このような長期に渡る開発で見積り期間通りに完了できたことは特筆に値する．年当たりに開発できた機種数(機種生産性)は過去の SPL を使わない例と比較すると 2.5 倍となった．また，工数見積りの精度としては見積りに対して実績が+2.7%であり，3%を切っている点で実用的な見積りといえる．また，コア資産の割合は 2 機種において 90%を超えており，ADM 手法を通してコア資産を大きく引き出すことができた．この開発後 4 年間で新分析装置 5 機種，新搬送路 2 機種接続に活用されており，形成されたコア資産の汎用性についても確認ができている．

7.4 節では，ADM 手法の組織面の効用を述べた．以下要点を箇条書きにする．

(1) コア資産開発とアプリケーション開発の調和：コア資産とアプリケーションのどちらにも改造がありうる前提で ADM 表は作られている．例えば一度決定したコア資産を中心に変更しない領域を広げる形では得られない調和が得られた．

(2) コア資産としての ADM 表：ADM 表を作成する過程で接続に対する変更パターンを学ぶことができている．ADM 表自体が将来に向けた変更点管理表にもなっている．

(3) ADM 手法見積りの準備：本事例においてはドメイン知識とソースコードとの関係を客観的に調査するのに 2 年を要している．ドメイン知識に関しては事前の準備があって初めて効果を得ることができる．

(4) 規制が求められる業界における活用：本論文における ADM 手法の対象は医用分析装置であるが，他の規制が求められる分野でも活用の可能性がある．

(5) コア資産チームの所属：独立なチームとせず，アプリケーション開発チーム内にアーキテクチャ要素単位に分散させて配置した．これにより Reactive 型開発を円滑にした．

(6) サブリーダの選出：サブリーダ選出にあたり，客観的な開発量，難易度を提示する上で ADM 表が役立った．

(7) WBS：技術者は大きくアーキテクチャ要素向け技術者，分析装置向け技術者，コア資産開発向け技術者に分かれるが，これらのコミュニケーションを助けた．

以上，アーキテクチャ要素をドメイン要素単位に分解してコア資産とアプリケーションの区分け解析を行う ADM(Architecture Domain Matrix)手法について述べた．この手法は AIMS(Analyzer Integration Management Software)のような長期で多岐に渡るソフト改造に対して生産効率を向上させることが示された．ADM 手法は，長期間に渡り変更の可能性が少ないアーキテクチャ構成要素とドメイン知識要素が存在するような製品開発を前提にしている．

ADM 手法特有の効果としては，コア資産を鳥瞰的に抽出しマクロ感をもって見積もれる点，将来に向けた組織的学習効果もある点，開発組織のコミュニケーションを円滑化する点があげられる．今後は，それを活かす他の分野での活用も期待される．

第８章　 適用製品

(1)　大型自動分析装置の開発

筆者は 1989 年から異種の自動分析装置を複数組み合わせた新しいモデルのシステム開発に従事し，開発の構想時点からその方式設計を行った．システムは，処理性能や分析方式の異なる異種自動分析装置を搬送ラインで結合して血液や尿などの検体の検査を自動化するものである．開発した方式はモジュール組合せ方式と呼び，1997年に発売された自動分析装置から適用を始めた．顧客のニーズに合わせて複数台の異種分析装置を搬送路に接続させ組合せる方式は，当時業界初であった．この方式の開発にあたり，筆者は本論文で述べたサイドトラック方式の制御方式を考案して適用している．

システムは，基本分析の検体を大量・効率的に分析できる生化学分析装置(D モジュールと呼ぶ)と検体数は少ないが多種項目を効率的に分析できる生化学分析装置(P モジュールと呼ぶ)を最大 4 台まで搬送路に結合できる構成であった. 当初の組合せとしては標準タイプとして 4 つの組合せを製品化した[40].

表 8‐1　1997 年発売のモジュール組合せ方式システムの構成

モジュール数	組合せ
2モジュール	PP, DP
3モジュール	DDP
4モジュール	DDPP

本論文で述べたサイドトラック方式については，前述したように 1996 年に国内特許[6]を，2003 年以降，米国及び欧州の特許[7][8]を取得している．方式を考案した時点のシステム構成図が図 8‐1 である．

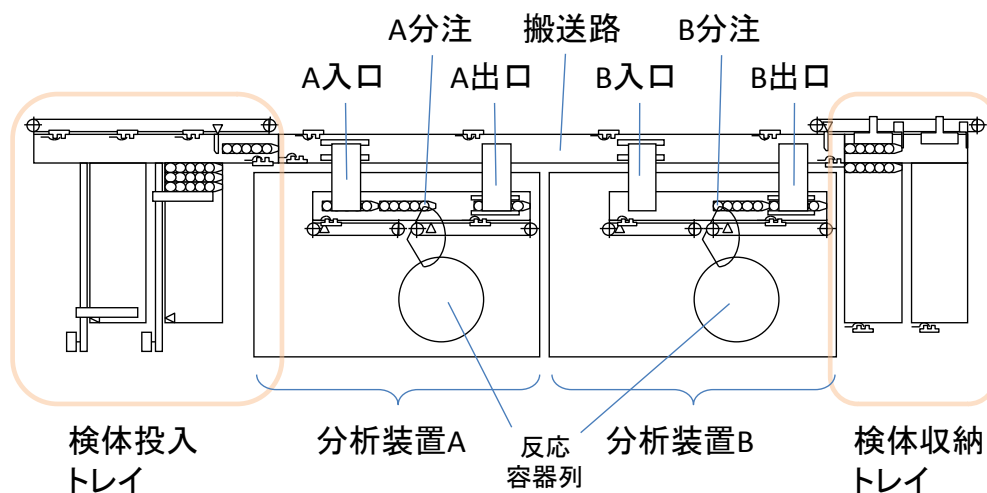



図　８-１　システム構成図

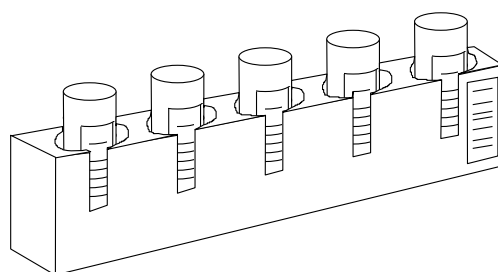



図　８-２　検体ラック

図　８-２には検体が入った試験管を保持するラックの概観形状が描かれている．ラックには試験管が５本支持され，ラック及び試験管それぞれにバーコードラベルが貼られている．この検体ラックが図　８-１の左にある投入トレイに整列され，レバーにより１ラックごと搬送路に押し出されていく．投入トレイは２列あり，一方のトレイのラックが無くなると，他方のトレイに切り替えてラックを押し出す．搬送路に入る直前にバーコードが読まれ，どの分析装置に行くかが決定されると検体は搬送路経由で分析装置の入口に引き込まれる．ラックの検体が反応容器に小分けされる分注が行われると次の装置に向かうために分析装置の出口バッファにラックが送られる．分析を終了したラックは図の右側にある収納トレイに収納されていく．投入トレイと同じく，収納トレイも２列ある．

このシステムが開発されるまでは，システム化を必要としている顧客に対しては，顧客ごとに搬送路を個別に設計して分析装置に合わせて接続する必要があったため多大なコストと開発時間がかかった．このシステムによりスペース効率の向上及びランニングコストの低減が実現され検査業務の効率化に貢献した．

この後，モジュール組合せ方式は受け継がれ，2002 年には生化学的検査に加えて免疫学的検査の分析装置(E モジュールと呼ぶ)を組合せることを可能にした[41]. これにより，計 3 種類(P, D, E)の分析装置を最大 4 台まで搬送路で結合して組合せることが可能となった．免疫学的検査とは，生体における抗原抗体反応を応用して微量成分を高感度に測定する検査で，例えば，腫瘍マーカー，ホルモン，感染症関連項目などの測定が行われる．

このようなシステム製品の特長は，顧客のニーズに合わせてシステム規模を拡張することができるのは当然ながら，加えて,『インテリジェント・ラック搬送技術による分析効率の向上』と紹介され，搬送機能が製品特長として大きくフィーチャされている[42]．検体搬送における渋滞を防ぎ追い越しを可能にした点をモジュール組合せ方式の利点として紹介している．

以上の製品は大規模病院および大規模臨床検査企業向けの大型装置である．この生化学と免疫の組合せにより, 全世界で大型装置の高いシェアを獲得することができた．

(2)　中型自動分析装置のモジュール組合せ方式適用とプラットフォーム開発

これに加えて 2006 年には中規模病院向けに中型生化学・免疫統合自動分析システムが開発された[43]. 生化学的検査の分析装置と免疫学的検査の分析装置の 2 種類を，モジュール組合せ方式により顧客施設の規模に合わせて構成することができる．このシステムの登場により，測定対象項目数は飛躍的に拡張された．2007 年時点で，生化学・免疫検査対象の 150 以上の検査項目を測定でき，1 台のシステムで生化学・免疫検査分野の検査ワークロードのおよそ 95%をカバーするに至った[1]．また，搬送路で分析装置を結合することにより検査のワークフローが統合され，サンプルの小分け分注回数は約 80%低減した[1].

また, このような異種自動分析装置を搬送ラインに結合する方式は更に進化を遂げ,

中型生化学・免疫統合自動分析システムの開発の後には生産性向上を目指してプラットフォームの開発が行われた．本論文で述べた ADM 手法はそこで考案され適用された．本文中でも述べたように，プラットフォームは 3 種の新分析装置を接続して製品開発に貢献した．その後 4 年間で 5 種の分析装置，2 種の搬送路を接続する実績を得ている．これらの装置も，世界市場でご利用いただいている．

株式会社日立ハイテクノロジーズ(以下，日立ハイテク)と Roche 社(本社：スイス，以下ロシュ社)は生化学・免疫自動分析装置を中心とした両社のコラボレーション事業を展開している[44]. この強固なパートナーシップは 36 年間に渡り継続されてきており，両社は世界中の医療機関に 55,000 台以上の生化学・免疫自動分析装置を納入している．両社は業界に先駆けたイノベーションを生み出しており，生化学・免疫統合型の装置はその一つである.『日立ハイテクの最先端技術を盛り込んだ信頼性と堅牢性に優れた装置と、ロシュ社の豊富な試薬メニュー及び世界最高レベルの新規試薬開発技術力という両社の強みをより効果的に発揮することで、今後もさらなる事業拡大が図れると期待しています。』(文献[44]より引用) とあるように，装置の最先端技術の一つとして，本論文で紹介した自動搬送制御技術及びそのシステム開発技術が位置付けられる．

更にホームページから引用するならば，『2006 年に市場投入した生化学・免疫統合型自動分析システムの貢献により，世界市場でのシェア（2009 年度当社推定）は，20%と非常に強いポジションを維持』(文献[45]より引用)している．世界市場で高い評価をいただいた背景には分析装置に関わる広範に渡る技術力や営業に関わる多くの関係者また提携会社の絶大なる努力がある．その技術要素の一つが，本論文で論述した自動搬送制御技術及びそのシステム開発技術であり，本論文の目的達成を裏付けている．

第９章　結論

９.１　結論

検体検査に関わる自動分析装置は 1970 年代から近年に至るまで大きな進化を遂げてきた．日立臨床検査自動分析装置は，1971 年に国産第 1 号機が出荷されその後多くの分析装置が開発されてきている．自動で分析できる検査項目の種類も桁違いに拡大され，現在は 100 を超えている．また，世界市場においても分析装置の優位性が認められている．このような分析装置個別に技術革新が起きる中，一方で，搬送路を装置内に内蔵して複数の内部分析装置を結合してシステム化する大型分析装置も処理性能を飛躍的に向上してきた．

本論文では，このシステム化の流れを支える基幹技術として，処理性能を高める搬送制御及びその開発を効率化する開発手法を提案した．

搬送制御に関する処理性能における課題は 2.1 節に述べた．まとめると以下のようになる．

- 【搬送制御方式】検体はシステム内の複数分析装置に立ち寄りながら分析されるが，この装置経路がばらつくことにより処理性能が低下することのないサイドトラック方式とその搬送制御方式を提案する．サイドトラック方式のハード構成は分析装置側に検体を引き込んで処理する引き込みラインを設けている．制御的には，引き込み部のバッファ及び検体の追い越し機能がばらつきに対して有効であることが期待される．この方式が旧来方式であるパイプライン方式に対して優位であることを以下のように解析的に明らかにして数値実験により検証する．

- 【限界搬送能力】サイドトラック方式の処理性能は搬送路の搬送能力に深く依存しているが，処理性能を引き出すために最低限必要となる搬送能力（限界搬送能力）を明らかにする．

- ■ 【サイドトラック方式の優位性】検体の装置経路がばらつくことに対してサイドトラック方式が処理性能を向上させているメカニズムとパイプライン方式に対する優位性を検体処理時間の圧縮率として数値的に明らかにする.

- ■ 【現行仕様を超えたときのサイドトラック方式の優位性】現行の製品仕様である分析装置台数，バッファ数を超えた構成において，サイドトラック方式の処理性能がパイプライン方式に対して依然として優位であることを数値的に明らかにして，今後のシステム設計に資する.

それぞれ掲げた搬送制御の課題に対して以下のような成果を得た.

- ✓ 【搬送制御方式】第 3 章においてサイドトラック方式の制御方式を提案した. 分析装置内ラインにおけるバッファの空きを集中管理することにより，検体のばらつきにより生じうるバッファの空きを抑制することができる．また，加えて分析装置に依存しない形で搬送を制御できる点でプラットフォームとして適している．この方式を以下に述べる数値実験で用いてサイドトラック方式の優位性を導いている.

- ✓ 【限界搬送能力】4.1 節においてサイドトラック方式の処理性能を引き出すために最低限必要となる搬送能力を解析的に明らかにした．各装置における検体処理時間を一定とすると，1 検体当たりの搬送時間が(検体処理時間)/((装置数)＋1)を超えるとシステム処理性能が大きく劣化することを導出した．つまり，システム処理性能を維持するための限界搬送能力を導いた. この解析結果を 5.1 節において装置台数を 2～4 台構成として数値実験で検証した．その結果，最大 4 台構成とすれば，検体処理時間の 1/5 以内であればシステム処理性能が劣化しないという結果が得られ，解析結果と一致した.

- ✓ 【サイドトラック方式の優位性】4.3 節においてサイドトラック方式の処理性能が検体の装置経路ばらつきに対して対処するメカニズムを解析的に明らかにした．ばらつきによって連続的に発生する空きの長さが確率分布を持ち，その分布を入力として空きが低減された分布を出力するフィルタとして搬送システムをモデル化した．このモデルは 5.3 節において数値実験で検証され，具体的

な処理性能を明らかにした．その結果，引き込みラインの入口出口それぞれにバッファを 1 個もつ装置 4 台構成において，ランダムな分析装置間経路をもつ検体列を検査処理終了するまでの時間は，STS が PLS に対して平均 30%圧縮するという結果を得た．これにより，従来漠然としていたサイドトラック方式の優位性が客観的数値的に明らかになった．

✓ 【現行仕様を超えたときのサイドトラック方式の優位性】5.4 節において，装置台数を現行の 4 から 8 まで，バッファ数を現行の 1 から 3 まで拡張したときのサイドトラック方式の優位性を数値的に明らかにした．その結果，STS のバッファ 2 個以上にしても更に 30%圧縮するほどではないことが明らかになった．4 台構成から 8 台構成に渡って，バッファ数が 2 となると圧縮率は 5 ポイントほど改善すること，従って，バッファ数は 1 つだけでも十分な圧縮率を達成できることが分かった．バッファ数は大きくなると装置コストに悪影響を与えるため，この結果は将来サイドトラック方式を改良設計するにあたり大いに有益な情報を与えている．

また，このような搬送システムを開発するにあたっての課題は 2.2 節に述べた．まとめると以下のようになる．

■ 【プラットフォームの必要性】複数の分析装置を統合するソフトウェアシステムにおいて，新しい分析装置を接続する毎に開発コストが多大にならないようにする工夫が求められる．このため分析装置を接続する毎に変わらないコア資産を抽出して維持するソフトウェア・プロダクトラインの適用を前提に，見積り方式としてアーキテクチャ要素とドメイン要素の両面からソースコードを解析する ADM 手法を提案した．これにより以下の効果を達成する．

■ 【見積りにおけるコア資産コスト統制】複数の分析装置を統合するシステム開発は，開発期間が長期に渡り，このため参考となる製品が少なく，過去の傾向からコア資産を導出することが難しい．適切なコア資産設計のための見積り方式が求められる．

■ 【見積り精度の向上】開発サイクルが長いためコア資産を抽出するための参考

ソースコードが一番最近開発された単一システム製品に限られる．このため単一のシステム製品でも十分な見積り精度を得られるための工夫が求められる．

- ■ 【長期に渡る開発への支援】見積り時点で，長期開発で課題となる人材配置及び開発プロセスへの支援が必要となる．

以上掲げた搬送制御システム構築上の課題に対して以下のような成果を得た．

- ✓ 【プラットフォームの必要性】ADM 手法を適用して 1.5 年に渡って 3 つの分析装置を搬送路に結合する統合システムの開発を行い，プラットフォームの基礎となるコア資産を確立した．ここで抽出されたコア資産は，その後 4 年間で新分析装置 5 機種，新搬送路 2 機種接続に活用されている．これは抽出されたコア資産が実効的であったことを示している．
- ✓ 【見積りにおけるコア資産コスト統制】上記プロジェクトにおいて，1.5 年で 3 機種の開発を完了させた．過去の自社内実績では SPL を使わない開発で 2 機種開発に 2.5 年をかけている．この 2 機種開発ともにその規模は,本論文における分析装置 A,B,C を平均した規模に比してほぼ対等といえる．従って，年当りの開発機種数は 0.8(機種/年)から 2.0(機種/年)と改善し，2.5 倍の生産性を得たといえる．これは ADM 手法によりアーキテクチャ全体に渡ってきめ細かくコア資産（不変部位）とアプリケーション（要求の変化に対応して変える部位）が分化され，特にコア資産の割合を高く維持できたことによる．これによってコア資産の開発コストを大幅に低減しながら開発を完了させることができた．
- ✓ 【見積り精度の向上】見積り工数に対して実績工数の見積り誤差は+2.7%であり実用的な見積り精度が得られたといえる．これは，ADM 手法において，コア資産とアプリケーションの分化とそれぞれの変更部位の特定が適切であったことによる．コア資産と決めた部位がアプリケーションにとって不都合になり，返って開発全体としては工数がかかってしまうことがある．このようなことを排除するために ADM 手法では全体像の中でコア資産候補，アプリケーション候補のいずれにも最適な変更をかけることができている．
- ✓ 【長期に渡る開発への支援】ADM 手法で生成された WBS が開発の中で有効に

活用された．特にアーキテクチャ単位に編成されたチームはそれぞれの専門領域において独立性を維持しながら，WBS を通じてレビュー，システムテストで全体の整合性をチェックすることができた．この点でコア資産を含む開発計画の困難さを低減できたといえる．

以上のように，本論文では，搬送システム化の基幹技術として提案した搬送制御及び開発手法において課題を全て解決できたと考える．

以上の搬送制御及び開発手法は実際の製品に適用された．自動で分析できる検査項目が増え，検体が経由する装置のばらつきが大きくなり，このような変化に対処すべく 1990 年代に新しい方式の大型自動分析装置が開発された．そこで生まれたモジュール組合せ方式と呼ばれる搬送システムには上述した搬送制御が採用されている．開発当初は 2 種の分析装置を組合せていたが，2000 年代に入り更に 1 種を加えて 3 種を組合せる大型分析装置に拡張された．このような拡張性は当時業界初であり，大型分析装置は世界市場で高いシェアを築いた．一方中型自動分析装置にもこの搬送システムの方式が導入され，他方，分析装置を接続する際の生産性向上を目的に上述の開発手法が適用されてプラットフォームが開発されるに至った．これらにより，自動分析装置の世界市場におけるシェアは高く維持されており，本論文の目的達成を裏付けている．

９.２　今後の研究課題

(1) 空きの確率分布の活用

本論文では，装置経路にばらつきをもつ検体を入力として検体搬送システムの処理性能を解析し，数値実験により検証した．このような離散的な物の流れを解析する研究領域として，第 2 章でも関連する研究として挙げたように，スケジュール問題がある．生産において部品が複数の機械を経由しながら最終製品に仕上がっていくプロセスが解析の対象となる．例えば本論文と同様に makespan が最小になるような部品のスケジュールを算出するアルゴリズムが研究される．

しかし，本論文では離散的な物の流れにおいて，物がある機械に送られることによって他の機械に空きが発生するという現象を捉えている．物の到着間隔は連続でもよ

い．連続に到着する物でも，ある機械に立ち寄らないような物がどれくらいあるかをその機械への空き確率分布として表現している．

この空きは，搬送システムのオンラインスケジュールによって圧縮される．スケジュールはある機械の空きを発見したら早期にそこを他の物で埋める．つまり，搬送システムは入力となる空きの確率分布を圧縮された空きの確率分布に変換するフィルタとしてモデル化されている．

このように本論文のモデルは，医療の検体搬送に留まらず，他のアプリケーションも探索できそうである．例えば，生産現場では部品の流れにおいてある機械への空きが生じる場合，それをどの程度埋めることができるかを圧縮率で示すことが可能となる．その圧縮率は当該生産システムの性能特性となる．

搬送のオンラインスケジュールシステムがもつシステム特性を圧縮率で示すことによって，客観的にシステムを評価することができる．検体搬送システム以外のアプリケーションを探索して適用する可能性が十分にあると考える．

(2) コア資産の変化パターン

本論文では，レガシーソフトウェアからコア資産を抽出するために，次に開発される新たな分析装置の仕様を参照して，変化の少ないものをコア資産に，変化の多いものをアプリケーションに色分けしている．コア資産とアプリケーションの違いは，端的にいえば変化頻度といえる．コア資産は長期に渡り不変であり，アプリケーションは要求仕様に変化があれば対応して変化して，外界の変化をクッションのように吸収する．このように考えるとコア資産とアプリケーションとの違いは相対的である．変更頻度が多ければアプリケーションで変更頻度が少なければコア資産となる．

企業活動の中でコア資産は製品に占める割合が大きくなること，逆にいえばアプリケーションが製品に占める割合が小さくなることが期待される．これは，外界の変化に対して低コストで対応できる製品であることを意味する．そのようなコア資産は，恐らく顧客の変化をある程度予見できているであろう．

しかし，将来に渡って絶対に変わらないコア資産はあり得ないと考える．コア資産は顧客の要求仕様における変化の鈍い部分を捉えているだけで，いつかはその要求仕

様も変化する．そのときコア資産はどのように変化するか，それが顧客のどのような要求仕様の変化に対応するかを研究することは，変化に対応するコア資産のあるべき構造を設計する上で重要となる．このようなコア資産の変化パターンについては追跡して調査することが望まれる．

(3) 組込みソフトにおける ADM 手法利用

ADM 手法はアーキテクチャ要素とドメイン要素がクロスするセルにソースコードを分割して，その変更要否を分析する手法である．本開発においては，ドメインとして分析装置のデータ管理ワークフローを採用した．医療機器においてデータ品質管理は重要であり，検査室では標準的な手順が決められている．どんな特殊な分析装置が導入されたとしても，検査室では標準的な手順によりデータを管理する必要がある．このためワークフローは安定であるとみなされ，よって，ワークフローはソースコードを分析するにあたってアーキテクチャの層構造と同様に安定なフレームを提供する．

このような安定的なワークフローは規制が求められる業務の世界で見つかる可能性があることを組織面の考察で述べた．しかし，一般の制御系の組込みソフトにおいては，状態遷移がこれに相当すると考える．特に状態に対応してプログラムが構築されているケースでは，状態要素とアーキテクチャ要素で囲まれたセルにプログラムを位置付けることにより，ADM と同様の解析ができると考える．

医用分析装置の分野に限らず，他の分野にも ADM 手法を適用し，その効果を研究対象としたい．

謝辞

本論文を書き上げてこれまで多くの方々が支えてくださったことを振り返ると感謝の気持ちで一杯になります．ここに皆さまへ感謝の言葉を述べたいと思います．

まず，研究を進めるにあたり直接ご指導をいただいた田野俊一教授に感謝いたします．論文初心者の私に対して論旨探索の道を開いていただき，ゼロから指導いただきました．

また，本論文をまとめるにあたり，貴重なご助言と支援をいただきました渡辺俊典名誉教授，舩橋誠壽氏，増位庄一氏に厚く御礼申し上げます．

審査を快く引き受けてくださいました大学院情報システム学研究科の阪口豊教授，末廣尚士教授，古賀久志准教授，田原康之准教授に感謝申し上げます．また，これまで論文を熟読いただき，誤りを見つけて頂いた情報メディア学講座事務員の岸本雅代氏に感謝申し上げます．

研究を始めるきっかけを与えていただいた日立オートモティブシステムズ株式会社森清三氏，株式会社日立製作所インフラシステム社中野利彦氏に感謝申し上げます．

本論文は日立自動分析装置の開発に深く関連しており，これまで装置を育てていただいた諸先輩には常に感謝の念を忘れえません．加えて，現在装置の開発に従事している開発者の皆さまには敬意を表するとともに感謝申し上げます．本学への入学を快く認めてくださった株式会社日立ハイテクノロジーズ松坂尚専務執行役に感謝申し上げます．

最後に，常に心の支えとなった妻和子，母みどり，父一成に感謝します．

関連論文

第 3 章，第 4 章，および第 5 章

兒玉隆一郎，島袋潤，高木由充，小泉忍，田野俊一：Architecture Domain Matrix 手法による医用分析装置統合システムソフトウェアの開発，情報処理学会論文誌，Vol. 55，No. 8，pp. 1796-1806 (2014).

Ryuichiro Kodama, Jun Shimabukuro, Yoshimitsu Takagi, Shinobu Koizumi, Shun'ichi Tano: Experiences with Commonality Control Procedures to Develop Clinical Instrument System, Proceedings, 18th International Software Product Line Conference (SPLC), pp. 254-263 (2014).

第 6 章および第 7 章

以下は 2014 年 9 月条件付採録

兒玉隆一郎，田野俊一：医用検体搬送システムの処理性能解析-検体経路のばらつきとの関係-，情報処理学会論文誌：数理モデル化と応用(TOM).

参考文献

1) 今井恭子:健康社会を支える生化学･免疫分析技術, 日立評論, Vol. 89, No. 12, pp. 958-959 (2007).

2) 経済産業省商務情報政策局　医療・福祉機器産業室：経済産業省における医療機器産業政策について　平成 25 年 9 月 6 日，経済産業省（オンライン），入手先〈http://www.jasis.jp/2013/pdf/result/130905_04_kakudo.pdf〉(参照 2014-09-15).

3) 亀井幸子：これからの生化学検査，医器学, Vol. 70, No. 2, pp. 62-67 (2000).

4) 高畑藤也，池田俊幸，松本孝一 ほか：検査の全自動化を目指した検体検査トータルシステム，日立評論，Vol. 73，No. 11，pp. 1003-1008 (1991).

5) 左藤猛英：血液生化学自動分析装置の開発，精密工学会誌, Vol. 61, No. 1, pp. 59-64 (1996).

6) 児玉隆一郎, 三巻弘, 三村智憲, 野田貴之：自動分析装置, 日本国特許第 3175729 号(2001).

7) R. Kodama, H. Mitsumaki, T. Mimura, T. Noda: Method of conveying sample rack and automated analyzer in which sample rack is conveyed, U. S. Patent 6,599,749, 2003-07-29.

8) R. Kodama, H. Mitsumaki, T. Mimura, T. Noda: Method of conveying sample rack and automated analyzer in which sample rack is conveyed, European Patent 0,801,308, 2006-01-18.

9) 鍵政豊彦：日立グループにおける組込みソフトウェア開発力強化の取組み, JEITA 組込み系ソフトウェア・ワークショップ(2009).

10) 桶野励，原田倫孝：複数台のコンベアベルトを組み合わせた搬送システムの構築，精密工学会誌，Vol. 78, No. 12, pp. 1105-1111 (2012).

11) 森村英典，大前義次：応用待ち行列理論（OR ライブラリー 13)，日科技連出版社（1975).

12) 牧野都治：待ち行列の応用，森北出版（2011).

13) 木瀬洋，関口恭毅：スケジューリング理論の基礎と応用-II：ジョブショップ問題とその計算複雑さ，システム制御情報学会誌 , Vol. 44, No. 10, pp. 601-608, (2000).

14) 今泉淳，森戸晋：需要を考慮した直列型有限バッファ待ち行列：近似的アプローチによるシステム挙動の分析，日本経営工学会誌，Vol. 44, No. 2, pp. 102-109 (1993).

15) ZAHNG Heng，石塚陽，山下英明 ほか：固定された加工順序および経路を持つジョブショップ型生産システムにおけるバッファ容量配分問題，日本経営工学会論文誌 Vol. 54, No. 1, pp. 11-18 (2003).

16) 岩田一明，大場史憲，室津義定 ほか：搬送およびバッファを考慮した大規模ジョブショップ・スケジューリング，日本機械学會論文集. C 編, Vol. 49, No. 437, pp. 133-141, (1983).

17) 山本久志，浜田康平，長塚 豪己：s 種類の結果を有する試行列における連の分布の効率的な算出方法：オーバーラップを許さない場合，日本経営工学会論文誌，Vol. 60, No. 4, pp. 218-225 (2009).

18) 三巻弘，児玉隆一郎，栗山裕也：臨床検査の効率的な運用に適したモジュール組合せ方式の血液自動分析装置，日立評論，Vol. 79，No. 10，pp. 757-762 (1997) .

19) 経済産業省：ＩＴ化の進展と我が国産業の競争力について，経済産業省（オンライン)，入手先〈http://www.meti.go.jp/committee/materials/downloadfiles/g70124b06j.pdf〉(参照 2013-09-08).

20) Software Engineering Institute, Carnegie Mellon University: Software Product Lines | Overview (online), available from 〈http://www.sei.cmu.edu/productlines/〉(accessed 2013-09-08).

21) 吉村健太郎，菊野亨：5 組込みシステムにおけるソフトウェアプロダクトラインの導入(<特集>ソフトウェア再利用の新しい波-広がりを見せるプロダクトライン型ソフトウェア開発-)，情報処理 Vol. 50，No. 4，pp. 295-302 (2009).

22) 吉村健太郎，ダルマリンガム ガネサン，ディルク ムーティック：プロダクトライン導入に向けたレガシーソフトウェアの共通性・可変性分析法，情報処理学会論文誌，Vol. 46，No. 8，pp. 2482-2491 (2005).

23) M. Fowler: Refactoring: Improving The Design of Existing Code, Addition Wesley, 1999.

24) 吉村健太郎，成沢文雄，菊野学：製品リリース履歴における論理的結合集合に基づいた横断フィーチャ分析法，情報処理学会論文誌，Vol. 50，No. 11，pp. 2654-2664 (2009).

25) K. Schmid: A comprehensive product line scoping approach and its

validation, Proc. the 24th International conference on software Engineering, ICSE '02, pp. 593-603, ACM (2002).

26) L. Northrop: Software product lines essentials (online), Software Engineering Institute, Carnegie Mellon University (2008), available from 〈http://www.sei.cmu.edu/library/assets/spl-essentials.pdf〉 (accessed 2014-02-04).

27) C. Stoermer, L. O'Brien: MAP – mining architectures for product line evaluations, Software Architecture, 2001. Proc. Working IEEE/IFIP Conference, pp. 35-44 (2001).

28) Knodel, J., John, I., Ganesan, D., et al: Asset Recovery and Their Incorporation into Product Lines, WCRE '05: Proc. 12th Working Conference on Reverse Engineering, Washington, DC, USA, IEEE Computer Society, pp. 120-129 (2005).

29) H. Koziolek, T. Coldschmidt, T. de Gooijer, et al: Experiences from Identifying Software Reuse Opportunities by Domain Analysis, Proc. 17th International Software Product Line Conference (SPLC) (2013).

30) Jones, L. G., and Bergey, J. K.: Exploring Acquisition Strategies for Adopting a Software Product Line, No. NPS-AM-10-041, CARNEGIE-MELLON UNIV COLORADO SPRINGS CO (2010).

31) 三巻弘，高橋克明，内藤茂昭 ほか：多目的用途に迅速に対応できる小形血液自動分析装置（医療の高度化・総合化に対応する医用機器・医療情報システム<特集>),日立評論, Vol. 73, No. 11, pp. 995-1002, (1991).

32) 森戸晋：モデルが見えるとき(<特集>モデリング-最適化モデリング-),オペレーションズ・リサーチ ： 経営の科学, Vol. 50, No. 4, pp. 225-228 (2005).

33) 大塚弘文，岩井善太，水本郁朗：：強正実性に基づく離散時間適応状態フィードバック制御とその液体搬送制御への応用，日本機械学會論文集. C 編, Vol. 63, No. 611, pp. 2296-2301 (1997).

34) R Core Team (2013): R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, URL 〈http://www.R-project.org/〉 (accessed 2014-08-19).

35) Software Engineering Institute, Carnegie Mellon University: A Framework for Software Product Line Practice, Version 5.0 (online), available from 〈http://www.sei.cmu.edu/productlines/frame_report/index.html〉 (accessed 2013-09-08).

36) Frakes, W.B. and Kang, K.C.: Software reuse research: Status and future, IEEE Transactions on Software Engineering, Vol. 31, No. 7, pp. 529-536 (2005).

37) Kang, K., Cohen, S., Hess, J., et al: Feature-Oriented Domain Analysis (FODA) Feasibility Study (CMU/SEI-90-TR-021, ADA235785) (online), Software Engineering Institute Carnegie Mellon University, available from 〈http://www.sei.cmu.edu/library/abstracts/reports/90tr021.cfm〉 (accessed 2013-09-08).

38) Hofman, P., Pohley, T., Bermann, A., et al: Domain Specific Feature Modeling for Software Product Lines, Proc. 16th International Software Product Line Conference (SPLC), pp. 229-238 (2012).

39) Takebe, Y., Fukaya, N., Chikahisa, M., Hanawa, T., and Shirai, O. Experiences with software product line engineering in product development oriented organization. in Proceedings of the 13th International Software Product Line Conference, Carnegie Mellon University, pp. 275-283 (2009).

40) 株式会社日立製作所：モジュール組合わせタイプの生化学自動分析装置を業界で初めて製品化－２種類の標準モジュールの組合わせにより、最適なシステムソリューションを提供－（オンライン），入手先 〈http://www.hitachi.co.jp/New/cnews/9704/0415.html〉 (参照 2014-10-01).

41) （株）日立ハイテクノロジーズ：「生化学分析と免疫分析の統合を可能にした臨床検査用自動分析装置」を発売（オンライン），入手先 〈http://www.hitachi-hitec.com/news_events/product/2002/nr020906.html〉 (参照 2014-10-01).

42) （株）日立ハイテクノロジーズ：モジュールアッセンブリ方式日立自動分析装置 7700 シリーズ（オンライン），入手先 〈http://www.hitachi-hitec.com/science/medical/7700series.html〉 (参照 2014-10-01).

43) 亘重範，神原克宏，浜地和弘，渋谷武志：中規模病院向け生化学・免疫統合自動分析システム,日立評論, Vol. 93, No. 3, pp. 40-45, (2011).

44) （株）日立ハイテクノロジーズ：ロシュ社との体外診断事業における提携契約を延長（オンライン），入手先 〈http://www.hitachi-hitec.com/news_events/ir/2014/nr20140409.html〉 (参照 2014-10-01).

45) （株）日立ハイテクノロジーズ：もっと日立ハイテク　製品紹介　ライフサイエンス（科学・医用システム）（オンライン），入手先 〈http://www.hitachi-hitec.com/ir/products_info/2008_2.html〉 (参照

2013-10-01).

46) 兒玉 隆一郎, 島袋 潤, 高木 由充, 小泉 忍, 田野 俊一 : Architecture Domain Matrix 手法による医用分析装置統合システムソフトウェアの開発, 情報処理学会論文誌, Vol.55, No.8, pp.1796-1806 (2014).

47) Ryuichiro Kodama, Jun Shimabukuro, Yoshimitsu Takagi, Shinobu Koizumi, Shun'ichi Tano: Experiences with Commonality Control Procedures to Develop Clinical Instrument System, Proceedings, 18th International Software Product Line Conference (SPLC), pp.254-263 (2014).

48) 兒玉 隆一郎, 田野 俊一 : 医用検体搬送システムの処理性能解析-検体経路のばらつきとの関係-, 情報処理学会 第 100 回数理モデル化と問題解決研究発表会 (2014 年 9 月 25 日).

49) 兒玉隆一郎, 田野 俊一 : 医用検体搬送システムの処理性能解析-検体経路のばらつきとの関係-, 情報処理学会論文誌 : 数理モデル化と応用(TOM) (条件付採録) (2014 年 9 月).

50) 及川雄大, 有我祐一, 近藤優丞 ほか : 台車及び NXT を用いた搬送系のペトリネットによる表現と解析, 計測自動制御学会東北支部 第 260 回研究集会, 資料番号 260-6 (2010).

51) 中西恒夫 , 久住憲嗣 , 福田晃 : 派生開発からプロダクトライン開発への漸次的移行プロセス XDDP4SPL におけるコア資産管理手法, 情報処理学会研究報告. EMB, 組込みシステム, 2013-EMB-28(9), pp. 1-6, (2013).

52) 上ヶ原誠, 浅野孝夫 : オンラインスケジューリングアルゴリズムの実験的性能評価, 情報処理学会研究報告. AL, アルゴリズム研究会報告, 2003(3), pp. 17-24, (2003).

著者略歴

兒玉　隆一郎（こだま　りゅういちろう）

1977 年 4 月	東京大学　理科一類入学
1981 年 3 月	東京大学　工学部　計数工学科卒業
1981 年 4 月	株式会社日立製作所入社
1988 年 4 月	Rochester Institute of Technology, College of Applied Science and Technology, Master of Science（米国）入学
1989 年 11 月	Rochester Institute of Technology, College of Applied Science and Technology, Master of Science（米国）卒業
2001 年 10 月	株式会社日立ハイテクノロジーズ入社
2013 年 4 月	電気通信大学　大学院情報システム学研究科　博士後期課程入学
2015 年 3 月	電気通信大学　大学院情報システム学研究科　博士後期課程修了